UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
CADERNOS DE MATEMÁTICA E ESTATÍSTICA
SÉRIE B: TRABALHO DE APOIO DIDÁTICO

INTRODUÇÃO AO CÁLCULO NUMÉRICO

ÁLVARO LUIZ DE BORTOLI
CAROLINA CARDOSO
MARIA PAULA GONÇALVES FACHIN
RUDNEI DIAS DA CUNHA

SÉRIE B, N° 59
PORTO ALEGRE, DEZEMBRO DE 2001

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE MATEMÁTICA PURA E APLICADA

Introdução ao Cálculo Numérico
2a. Edição

Álvaro Luiz de Bortoli
Carolina Cardoso
Maria Paula Gonçalves Fachin
Rudnei Dias da Cunha

Porto Alegre, abril de 2003.

Álvaro Luiz de Bortoli é professor adjunto da UFRGS, desempenhando suas atividades junto ao Departamento de Matemática Pura e Aplicada, do Instituto de Matemática, desde 1996. É formado em Engenharia Mecânica pela UFRGS (1987); Mestre em Engenharia Mecânica pela UFSC (1990); e Doutor em Aerodinâmica e Aeroelasticidade pela UFSC e *Deutsches Luft- und Raumfahrt Institut*, Alemanha (1995).

Carolina Cardoso é Bacharel em Matemática (ênfase Matemática Aplicada e Computacional) pela UFRGS (1998) e Mestre em Matemática Aplicada pela UFRGS (2001).

Maria Paula Gonçalves Fachin é professora adjunta da UFRGS, desempenhando suas atividades junto ao Departamento de Matemática Pura e Aplicada, do Instituto de Matemática, desde 1990. É Licenciada em Matemática pela UFRGS (1986); Mestre em Matemática pela UFRGS (1990); e *Doctor of Philosophy in Computer Science* pela *University of Kent at Canterbury*, Reino Unido (1994).

Rudnei Dias da Cunha é professor adjunto da UFRGS, desempenhando suas atividades junto ao Departamento de Matemática Pura e Aplicada, do Instituto de Matemática, desde 1994. É Bacharel em Ciências de Computação pela UFRGS (1988) e *Doctor of Philosophy in Computer Science* pela *University of Kent at Canterbury*, Reino Unido (1992). Exerceu as funções de programador de computadores e analista de sistemas no Centro de Processamento de Dados da UFRGS (1983-1994). Foi coordenador do Programa de Pós-Graduação em Matemática Aplicada da UFRGS (1999-2000) e atualmente ocupa o cargo de Vice-Diretor do Instituto de Matemática da UFRGS.

Sumário

Capítulo 1

Aritmética no Computador

1.1 Introdução

Hoje em dia, os computadores utilizam um sistema de numeração em base 2, em sua grande maioria. Esse sistema é chamado de *binário* e utiliza os algarismos 0 e 1 para representar os números (apesar de que quaisquer outros dois símbolos poderiam ser usados).

A nossa sociedade, ao contrário, utiliza um sistema de numeração *decimal*, ou base 10; muito provavelmente, pelo fato dos seres humanos terem dez dedos nas mãos, os quais eram utilizados – como uma criança os utiliza – para contar quantidades. A palavra *dígito*, sinônimo para algarismo, vem do latim *"digitus"*, dedo.

Ambos os sistemas citados – decimal e binário – são sistemas *posicionais*, i.e., os números são formados por somas de potências, convenientemente multiplicadas pelos algarismos. Por exemplo, o número

$$(420,325)_{10} = 4 \times 10^2 + 2 \times 10^1 + 0 \times 10^0 + 3 \times 10^{-1} + 2 \times 10^{-1} + 5 \times 10^{-3} \qquad (1.1)$$

é representado no sistema decimal como a soma das potências de 10 mostradas acima.

A principal característica de um sistema de numeração posicional é a necessidade da representação do *zero* por um símbolo. Aparentemente, o zero já era utilizado pelos maias e pelos babilônios, esses por volta de 300 A.C. O nosso sistema de numeração decimal foi inventado na Índia por volta do ano 600 D.C. e, tendo sido usado por muitos séculos pelos povos árabes no Oriente Médio, foi introduzido na Europa durante as invasões mouras, no período entre 1200 e 1600 (daí o nome de algarismos "arábicos").

Veja como o zero é importante num sistema posicional, comparado com um sistema não-posicional, como o romano, por exemplo. Nesse último, o número 401 é representado como CCCCI (os romanos não utilizavam as abreviações como IV para representar 4). Porém, no sistema decimal, o 0 é necessário para distinguir 401 de 41 – ele efetivamente serve como um "espaçador" dos algarismos, em termos das potências de 10.

É interessante notar que o sistema decimal era utilizado para representar apenas números inteiros, e não frações decimais, até o século XVII. Em países de língua inglesa, até hoje persiste o uso de frações inteiras como 1/4, 1/8, 1/16, 3/4, como por exemplo em placas de sinalização rodoviária e na especificação dos diâmetros de ferramentas.

Como dissemos ao iniciarmos esse capítulo, os computadores utilizam normalmente um sistema de numeração binário, ou base 2. Esse sistema não é, no entanto, tão recente quanto os computadores; na verdade, já era usado como base para um algoritmo de multiplicação no *Papiro Matemático de Rhind*, escrito há 4,000 anos atrás [12, pág. 7].

1.2 Representação em binário e decimal

Genericamente, podemos dizer que um sistema de numeração numa base β admite apenas os dígitos $0, 1, \ldots, \beta - 1$. Assim, o número $(1001,11101)_2$ representa o número $(9,90625)_{10}$, onde os subscritos indicam a base do sistema de numeração utilizado:

$$\begin{aligned}(1001,11101)_2 &= 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0 + \\ &\quad 1\times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3} + 0\times 2^{-4} + 1\times 2^{-5} \\ &= 8 + 0 + 0 + 1 + 0{,}5 + 0{,}25 + 0{,}125 + 0 + 0{,}03125 \\ &= (9,90625)_{10}\end{aligned}$$

Assim como existem números reais, em decimal, que tem parte fracionária com número infinito de dígitos – ditos *irracionais* – também existem números reais em binário com a mesma característica. Mais ainda, existem números reais, em decimal, cuja parte fracionária tem um número finito de dígitos, para os quais a sua representação em binário apresenta um número *infinito* de dígitos. Por exemplo, o número 1/10 não tem uma representação binária finita:

$$\frac{1}{10} = (0,0001100110011\ldots)_2 = \frac{1}{16} + \frac{1}{32} + \frac{0}{64} + \frac{0}{128} + \frac{1}{256} + \frac{1}{512} + \frac{0}{1024} + \ldots$$

mas o conjunto de dígitos 0011 repete-se.

1.2.1 *Bits*, *bytes* e palavras

A memória de um computador pode ser descrita como um conjunto de *palavras*. A maioria dos computadores tem sua memória estruturada de tal forma que cada acesso – de leitura ou escrita – é feito em termos de uma ou mais palavras, as quais são acessadas por um *endereço* único. Tipicamente, uma palavra é composta por 32 *bits*; processadores de última geração para microcomputadores pessoais com palavras de 64 *bits* já são uma realidade hoje.

Um *bit* (contração em inglês de "binary digit") é a menor unidade de informação armazenada em um computador, podendo representar os valores 0 ou 1. Um conjunto de 8 bits é chamado de *byte*; nele, podemos armazenar $2^8 = 256$ diferentes valores inteiros, através das combinações de 0 e 1 entre os diferentes *bits*.

1.2.2 Conversão entre representações

É conveniente saber como converter um número decimal para sua representação em binário e vice-versa. Em algumas aplicações envolvendo o uso de computadores, é necessário saber como converter para decimal um valor armazenado de forma binária.

Podemos efetuar a conversão de um número real decimal x para binário convertendo separadamente as partes inteira e fracionária de x – $ip(x)$ e $fp(x)$ – e depois justapor a representação binária dessas duas partes, separando-as por um ponto. Nos algoritmos apresentados a seguir, a conversão de decimal para binário resulta em um "string" de caracteres 0 e 1, e não num número formado pelos mesmos algarismos, pois não são representações equivalentes.

Suponha então o número $x = (401,640625)_{10}$. A representação binária de $ip(x) = (401)_{10}$ é obtida, inicialmente, dividindo-se 401 por 2; essa divisão devolve um quociente e um resto. O resto é, necessariamente, 0 ou 1. Após, divide-se esse quociente por 2, obtendo-se um outro quociente e resto. Esse processo é repetido até que o quociente seja 1; a representação binária é formada, então, pelo último quociente e pelos restos, tomados na ordem inversa a que foram obtidos. A tabela 1.1 mostra esse processo. A representação em binário de $ip(x) = (401)_{10}$ é, portanto,

$$\begin{aligned}(401)_{10} &= (110010001)_2 = \\ &= 1\times 2^8 + 1\times 2^7 + 0\times 2^6 + 0\times 2^5 + 1\times 2^4 + 0\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0 = \\ &= 256 + 128 + 16 + 1 = 401\end{aligned}$$

dividendo	quociente	resto
401	200	$\underline{1}$
200	100	$\underline{0}$
100	50	$\underline{0}$
50	25	$\underline{0}$
25	12	$\underline{1}$
12	6	$\underline{0}$
6	3	$\underline{0}$
3	$\underline{1}$	$\underline{1}$

Tabela 1.1: *Processo de conversão para binário da parte inteira de* $(401,640625)_{10}$*; os dígitos sublinhados comporão a representação binária.*

Para convertermos a parte fracionária $fp(x) = (0,640625)_{10}$, fazemos um processo de *multiplicações sucessivas por* 2. Inicialmente, multiplicamos $0,640625$ por 2, resultando em $1,28125$. O dígito à esquerda do ponto decimal será um dos dígitos da representação binária de $fp(x)$; como esse dígito é igual a 1, subtraímos 1 do número, resultando em $0,28125$. Esse número é, novamente, multiplicado por 2, resultando em $0,5625$; como o dígito à esquerda do ponto decimal é o 0, basta multiplicar novamente esse número por 2. O processo continua até que o número multiplicando seja igual a $1,0$; os dígitos 0 e 1, à esquerda do ponto decimal, formam a representação binária de $fp(x)$, agora na *mesma* ordem em que foram obtidos, conforme mostrado na tabela 1.2

multiplicando	resultado
$0,640625$	$\underline{1},28125$
$0,28125$	$\underline{0},5625$
$0,5625$	$\underline{1},125$
$0,125$	$\underline{0},25$
$0,25$	$\underline{0},5$
$0,5$	$\underline{1},0$

Tabela 1.2: *Processo de conversão para binário da parte fracionária de* $(401,640625)_{10}$*; os dígitos sublinhados comporão a representação binária.*

Logo, a representação em binário de $fp(x) = 0,640625$ é

$$\begin{aligned}(0,640625)_{10} &= (0,101001)_2 = \\ &= 1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}+0\times 2^{-4}+0\times 2^{-5}+1\times 2^{-6} = \\ &= 0,5+0,125+0,15625 = 0,640625\end{aligned}$$

e, portanto, podemos justapor as representações binárias de $ip(401,640625) = (110010001)_2$ e $fp(401,640625) = (0,101001)_2$, obtendo a representação binária de $(401,640625)_{10}$, a qual é $(110010001,101001)_2$.

Os algoritmos apresentados a seguir sumarizam os processos de conversão de decimal para binário e de binário para decimal.

Algoritmo 1.2.1 *Conversão decimal para binário*

```
proc conv_dec_para_bin(input: x; output: b)
    % x eh um numero real decimal em modulo e b eh a sua
    % representacao binaria, armazenada como um
    % "string" de caracteres.
    x ← |x|
    b ← conv_dec_para_bin_ip(ip(x))|conv_dec_para_bin_fp(fp(x))
endproc
proc conv_dec_para_bin_ip(input: x; output: b)
    % x eh um numero inteiro e b eh a sua representacao
    % binaria, armazenada como um "string" de caracteres.
    d ← ⌊x/2⌋
    r ← x − 2d
    x ← d
    b ← num2str(r)
    while x ≥ 1
        d ← ⌊x/2⌋
        r ← x − 2d
        x ← d
        b ← num2str(r)|b
    endwhile
endproc
proc conv_dec_para_bin_fp(input: x; output: b)
    % x eh um numero fracionario menor do que 1 e
    % b eh a sua representacao binaria, armazenada
    % como um "string" de caracteres.
    x ← |x|
    b ← '.'
    while x < 1
        d ← x * 2
        i ← ⌊d⌋
        if d > 1 then
          x ← d − 1
        else
          x ← d
        endif
        b ← b|num2str(i)
    endwhile
endproc
```

Algoritmo 1.2.2 *Conversão binário para decimal*

```
proc conv_bin_para_dec(input: b; output: x)
    % b eh um numero binario, armazenado como
    % um "string" de caracteres e x eh a sua
    % representacao em decimal.
    l ← length(b)
    p ← findstr(b,'.') % Localiza onde estah o ponto decimal em b
    if p ≠ 0 then % b eh um numero binario com parte fracionaria
      x ←conv_bin_para_dec_ip(substr(b, 1, p − 1))+
          conv_bin_para_dec_fp(substr(b, p, l))
    else % b eh um numero inteiro
      x ← conv_bin_para_dec_ip(b)
endproc
proc conv_bin_para_dec_ip(input: b; output: x)
    % b eh um numero binario inteiro, armazenado como
    % um "string" de caracteres e x eh a sua
    % representacao em decimal.
    l ← length(b)
    x ← 0
    k ← l
    for i = 1, 2, ..., l
      x ← x + str2num(b[i]) * 2^k
      k ← k − 1
    endfor
endproc
proc conv_bin_para_dec_fp(input: b; output: x)
    % b eh um numero binario menor do que 1, armazenado
    % como um "string" de caracteres (com '.' aa frente) e
    % x eh a sua representacao em decimal.
    l ← length(b)
    x ← 0
    k ← 1
    for i = 2, 3, ..., l % Desconsidera o 1o. caracter ('.')
      x ← x + str2num(b[i]) * 2^−k
      k ← k + 1
    endfor
endproc
```

Note, no entanto, que se por um lado um número inteiro decimal pode ser facilmente representado por um número inteiro binário, a representação da parte fracionária nesse sistema de numeração exige, normalmente, um número bastante elevado de dígitos. Por exemplo, $(0,249)_{10}$ – o qual difere de $(0,25)_{10} = (0,01)_2$ por apenas $(0,001)_{10}$ – não tem representação binária finita (pois essa diferença não pode ser representada de forma finita): seus primeiros 20 dígitos são

$$(0,00111111101111100011100\ldots)_2$$

e veja que, como $0,249 < 0,25$, necessariamente o primeiro dígito não-nulo na representação binária encontra-se à partir da terceira casa binária (pois $(0,25)_{10} = (0,01)_2$).

1.3 Representação de números em um computador

A representação de números em um computador está intimamente relacionada às operações que serão efetuadas com eles. Inicialmente, consideraremos a representação dos números inteiros e,

depois, passaremos aos números reais.

1.3.1 Representação de números inteiros

Hoje em dia, os números inteiros são armazenados, tipicamente, em uma palavra de 32 *bits*. Considerando apenas números *positivos*, temos um total de 2^{32} números possíveis de ser representados em uma palavra desse comprimento: 0, 1, ..., $2^{32}-2$, $2^{32}-1$. No entanto, é necessário que se manipule números inteiros negativos, e, nesse caso, devemos analisar as possibilidades existentes.

O sinal (+/−) é uma quantidade binária e, portanto, podemos armazenar essa informação em um único *bit*. Isso nos leva, portanto, a pensarmos numa representação chamada *sinal-e-módulo*: dos 32 *bits* de que dispomos, reservamos um para o sinal, e os restantes 31 representarão o valor absoluto do número. Essa representação apresenta duas carateristicas:

1. A perda de um *bit* implica na redução do intervalo de representação dos números. Agora, só podemos representar os números 0, 1, ..., $2^{31}-1$ (em módulo);

2. O número *zero* tem duas representações: +0 e −0.

Suponha, por exemplo, que quiséssemos efetuar +13 (−) 13; o resultado seria +0 ou −0? Além disso, seria necessário existir um circuito, dentro do processador, específico para se efetuar uma subtração; não seria melhor que a subtração fosse tratada como a soma de um número positivo e outro negativo?

Essa última característica é que leva ao uso de uma outra representação para números binários inteiros com sinal, chamada de *complemento-de-2*. Um número $-x$ em complemento-de-2 é obtido invertendo-se os *bits* da representação binária de $|x|$ e somando $(1)_2$ ao *bit* menos significativo. Para um conjunto de n *bits*, o intervalo de representação de números em complemento-de-2 é $-2^{n-1} \leq x \leq 2^{n-1}-1$ (ao passo que, em sinal-e-módulo, é $-(2^{n-1}-1) \leq x \leq 2^{n-1}-1$, pois um *bit* é usado para guardar o sinal de x). A tabela 1.3 mostra os inteiros representados em complemento-de-2, bem como em sinal-e-módulo, para $n = 3$.

	sinal-e-módulo	complemento-de-2
+3	$(011)_2$	$(011)_2$
+2	$(010)_2$	$(010)_2$
+1	$(001)_2$	$(001)_2$
+0	$(000)_2$	$(000)_2$
−0	$(100)_2$	–
−1	$(101)_2$	$(111)_2$
−2	$(110)_2$	$(110)_2$
−3	$(111)_2$	$(101)_2$
−4	–	$(100)_2$

Tabela 1.3: *Inteiros em sinal-e-módulo e complemento-de-2, para* $n = 3$ bits*; o* bit *mais à esquerda representa o sinal no formato sinal-e-magnitude.*

Em complemento-de-2, e usando uma palavra de 32 *bits*, um número *positivo* x satisfaz $0 \leq x \leq 2^{31}-1$, e sua representação em binário (cf. visto na seção anterior) é utilizada para armazená-lo, sem modificações. No entanto, um número *negativo* $-y$ é armazenado como a representação binária do número *positivo* $2^{32}-y$, e $-y$ é tal que ele satisfaz $1 \leq y \leq 2^{31}$.

Suponha, novamente, a operação +13(−)13. O número $(+13)_{10}$ tem a seguinte representação binária

$$(+13)_{10} = (00000000000000000000000000001101)_2$$

e $(-13)_{10}$, em complemento-de-2, é escrito como:

$$(-13)_{10} = (11111111111111111111111111110011)_2$$

Agora, para somarmos os dois, seguimos as mesmas regras da adição em decimal; procedemos do *bit* menos significativo para o mais significativo, somando dois *bits*, x e y, a cada vez. Por exemplo, $(1)_2 + (1)_2 = (10)_2 = (2)_{10}$, como esperado[1]. No entanto, é mais costumeiro tratar o resultado dessa soma como um *bit* de soma, s, e um de transporte c (do inglês "carry"), de acordo com a tabela 1.4.

x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabela 1.4: *Soma de dois* bits.

Note que os *bits* s e c são expressos em termos das operações lógicas $\oplus$ (*ou-exclusivo*) e $\wedge$ (*multiplicação lógica*): $s = x \oplus y$ e $c = x \wedge y$.

Retornando ao nosso exemplo, somamos $(+13)_{10}$ e $(-13)_{10}$:

$$\begin{aligned} & (\,00000000000000000000000000001101)_2 \\ + \;& (\,11111111111111111111111111110011)_2 \\ = \;& (\underline{1}00000000000000000000000000000000)_2 \end{aligned}$$

Veja que, a menos do *bit* mais significativo do resultado (sublinhado), obtivemos o valor esperado, ao somar um número e seu simétrico: zero. O *bit* mais significativo é apenas o *bit* de transporte, o qual é descartado, nesse caso.

Ao somarmos dois inteiros positivos, é possível que ocorra um "overflow", i.e. o resultado não pode ser representado na palavra (por exemplo, de 32 *bits*). O mesmo pode acontecer se somarmos dois inteiros negativos. No entanto, se somarmos dois inteiros de sinais opostos, não pode ocorrer "overflow". Se considerarmos a operação $x + (-y)$, e $0 \leq x \leq 2^{31} - 1$ e $0 \leq y \leq 2^{31}$, o resultado poderá ser sempre representado na palavra. Suponha agora que somemos os complementos-de-2 de x e $-y$, i.e. os números não-negativos x e $2^{32} - y$; isso resulta em $2^{32} + x - y = 2^{32} - (y - x)$. Se $x \geq y$, um *bit* de transporte – correspondente a 2^{32} – terá o valor 1, mas esse pode ser descartado, conforme visto anteriormente. Se $x < y$, o resultado cabe em 32 *bits*, e ele representa $-(y - x)$ em complemento-de-2.

1.3.2 Representação de números reais

Na representação de números reais, temos basicamente três alternativas: representação *racional*, por *ponto-fixo* e por *ponto-flutuante*.

1.3.2.1 Representação racional de números reais

A representação racional utiliza dois inteiros – numerador e denominador – para representar a fração. Apesar de permitir representar de forma exata as frações, as operações aritméticas envolvendo tal representação tornam-se bastante complicadas. Em sistemas de computação simbólica, como o MAPLE, MATHEMATICA e DERIVE, dentre outros, essa representação é adotada, porém cabe ressaltar que ela não existe em termos de "hardware"; esses sistemas executam por "software" as operações aritméticas envolvendo frações (incorrendo em um considerável aumento no tempo tomado para se efetuar as operações).

[1] Esse procedimento é o mesmo que ocorre no sistema decimal: o "vai-um" na soma de dois dígitos decimais indica que ela excedeu a 9 e, portanto, deve ser transportado o complemento de 10 da soma (indicado pelo transporte do 1) para o dígito subseqüente.

1.3.2.2 Representação de números reais em ponto-fixo

A segunda alternativa é chamada de ponto-fixo. Nesse caso, o ponto binário ocupa uma posição fixa (daí o nome) – existe uma quantidade pré-definida de dígitos binários à esquerda e à direita do ponto. A palavra do computador é dividida em três *campos*:

1. s, sinal do número ($|s| = 1$ *bit*);
2. e, dígitos à esquerda do ponto binário ($|e| = 15$ *bits*, por exemplo);
3. d, dígitos à direita do ponto binário ($|d| = 16$ *bits*, por exemplo).

Por exemplo, o número $-11,75$ é representado em ponto-fixo como

1	000000000001011	1100000000000000

Novamente, aqui, existem duas representações para o zero; porém, o principal problema reside no fato de que o intervalo de representação dos números é bastante pequeno, conforme veremos a seguir.

1.3.2.3 Representação de números reais em ponto-flutuante

A terceira maneira de representar números reais em um computador é chamada de ponto-flutuante. Ela é baseada na *notação científica*, i.e. um número real x em base decimal é expresso na forma

$$x = \pm M \times 10^{\pm E} \tag{1.2}$$

onde M é a *mantissa* e E é o *expoente*. Note que se exigirmos que S seja um número que satisfaça

$$\frac{1}{10} \leq M < 1$$

então podemos imaginar que o ponto decimal é movido à esquerda ou direita, ajustando-se convenientemente o valor de E – daí o nome "ponto-flutuante". Nesse caso, dizemos que o número x encontra-se em *notação científica normalizada*, pois o primeiro dígito após o ponto decimal é diferente de zero. A notação normalizada apresenta uma restrição, a qual é a impossibilidade de se representar o $x = 0$; essa restrição será removida mais adiante.

A notação científica normalizada pode ser facilmente estendida para números reais em base binária; nesse caso, temos

$$x = \pm M \times 2^{\pm E}, \quad \frac{1}{2} \leq M < 1 \tag{1.3}$$

de onde M é um número na forma

$$M = (0, b_0 b_1 b_2 b_3 \ldots)_2, \quad b_0 = 1$$

Por exemplo, o número $-11,75$ pode ser representado como

$$(-11,75)_{10} = (-0,101111)_2 \times 2^{(+100)_2}$$

Novamente, aqui, o número 0 não pode ser representado, pois $M \geq 1/2$, por definição; a representação do 0 deve ser tratada, portanto, como um caso especial.

Agora, observando a equação (1.3), podemos ver que para representar o número x naquela forma, devemos armazenar em uma palavra quatro informações distintas: o sinal da mantissa, a mantissa, o sinal do expoente e o expoente. Esses dois últimos podem ser representados separadamente ou simultaneamente; nesse caso, pode-se usar complemento-de-2 (apesar dessa forma não ser utilizada usualmente) ou deslocamento ("biased exponent", como no padrão IEEE-754).

Também podemos observar que, como $b_0 = 1$ em M, não é necessário representá-lo; isso nos permitirá economizar um *bit* da palavra que armazenará x.

Essa representação, também chamada de *sistema de ponto-flutuante*, é denotada por

$$F = (\beta, |M|, |E|) \tag{1.4}$$

onde β é a base na qual os números estão expressos, $|M|$ e $|E|$ são a quantidade de dígitos utilizados para representar a mantissa e o expoente.

Para fins ilustrativos, vamos considerar uma palavra de 32 *bits*, dividindo-a em três campos:

1. s, sinal do número ($|s| = 1$ *bit*);
2. E, o expoente do número, expresso em complemento-de-2 ($|E| = 8$ *bits*, por exemplo);
3. M, a mantissa do número, expressa na forma $(0, 1b_1b_2b_3\ldots)_2$ ($|D| = 23$ *bits*, por exemplo);

Nesse caso, o número $(-11,75)_{10} = (-0,101111)_2 \times 2^{(+100)_2}$ será expresso como

1	00000100	10111100000000000000000

1.3.2.4 Tratamento do zero

Num sistema de ponto-flutuante em notação científica normalizada, a representação do zero é um caso especial, pois qualquer número x nesse sistema é tal que sua mantissa é um número $M > 0$. Note que o padrão de *bits*

0	00000000	00000000000000000000000

não representa 0, mas sim 1 (uma vez que b_0 não é armazenado).

Temos, então, duas opções para representar o zero:

1. Representar explicitamente b_0: com isso, reduzimos a precisão, pois o *bit* b_{23} da mantissa não poderá ser representado;
2. Escolher um certo valor de E o qual, quando o padrão de *bits* de M for $00\ldots00$, será considerado como representando o 0. Essa é a estratégia utilizada no padrão IEEE-754,

Note que, em ambas opções, persiste a representação dupla para o zero, $+0$ e -0; usualmente o sinal é desconsiderado, nessa situação.

1.3.3 Caracterização de uma representação

A fim de caracterizarmos uma representação de números reais, seja em ponto-fixo ou ponto-flutuante, podemos definir algumas quantidades, as quais são:

1. A precisão, p, é a quantidade de *bits* disponível para representar o número;
2. O menor número representável, em módulo, `MINR`;
3. O maior número representável, em módulo, `MAXR`;
4. O menor número representável, ε, tal que $1 + \varepsilon \neq 1$, também chamado de *epsilon da máquina* ou *unidade de arredondamento da máquina*;
5. A menor separação possível entre dois números representáveis, `ULP` (do inglês "units-in-the-last-place").

Para um sistema de ponto-fixo, teremos então

1. $p = |e| + |d|$;
2. `MINR` é obtido fazendo-se $s = 0$, $e = 0$ e colocando 1 no *bit* menos significativo de d;

3. MAXR é obtido fazendo-se $s = 0$, e e e d tendo todos os seus *bits* iguais a 1;

4. $\varepsilon \equiv$MINR;

5. ULP$\equiv$MINR.

Usando-se uma palavra de 32 bits, dividida em campos e com 15 *bits* e d com 16 *bits*, podemos calcular essas quantidades, conforme mostra a tabela 1.5.

p	31
MINR	$2^{-16} \approx 0,000015$
MAXR	$(2^{15}-1)+\sum_{i=1}^{16} 2^{-i} = 32767,9999847412109375 \approx 2^{15}$
ε	2^{-16}
ULP	2^{-16}

Tabela 1.5: *Valores caracterizadores de uma representação em ponto-fixo.*

Já para um sistema de ponto-flutuante, essas quantidades são obtidas de forma diferente:

1. $p = |M| + 1$ (pois $b_0 = 1$ não é armazenado);

2. MINR, é obtido fazendo-se $s = 0$, $E = -128$ e $M = 1/2$.

3. MAXR, é obtido fazendo-se $s = 0$, $E = 127$ e M tendo todos os seus *bits* iguais a 1;

4. $\varepsilon = 2^{-(p-1)}$, para arredondamento por corte, ou $\varepsilon = \frac{1}{2}2^{-(p-1)} = 2^{-p}$, para arredondamento por adição (ver 1.3.4);

5. ULP$= (0,00\ldots01)_2 \times 2^E = 2^{-(p-1)} \times 2^E = \varepsilon \times 2^E$, para qualquer número x na forma $\pm M \times 2^{\pm E}$.

Usando uma palavra de 32 *bits* dividida conforme expresso acima, essas quantidades tem os seguintes valores, conforme mostra a tabela 1.6, onde ε foi calculado usando-se arredondamento por corte. Comparando com a tabela 1.5, é fácil notar que a representação em ponto-flutuante

p	24
MINR	$2^{-1} \times 2^{-128} \approx 0,146937 \times 10^{-38}$
MAXR	$(\sum_{i=1}^{24} 2^{-i}) \times 2^{127} \approx 0,170141 \times 10^{39}$
ε	$2^{-(24-1)} = 0,119209 \times 10^{-6}$
ULP	$\varepsilon \times 2^E = 2^{-23+E}$

Tabela 1.6: *Valores caracterizadores de uma representação em ponto-flutuante.*

oferece um intervalo muito maior de números representáveis; além disso, a separação entre esses números é bem menor.

Na representação em ponto-flutuante, é importante estabelecer o intervalo de valores possíveis para o expoente. Os limites desse intervalo são o menor e o maior expoente, MINE e MAXE, respectivamente, e sua definição depende de como os expoentes são armazenados, conforme a tabela 1.7.

Cabe, aqui, uma observação referente ao ε. Suponha que se desconheçam as características do sistema de ponto-flutuante de um computador ou calculadora; nesse caso, é possível estimar o ε, usando o algoritmo 1.3.1, o qual baseia-se na definição $1 + \varepsilon \neq 1$:

	sinal-e-módulo	complemento-de-2
MINE	$-\beta^{\lvert E\rvert-1} - 1$	$-\beta^{\lvert E\rvert-1}$
MAXE	$+\beta^{\lvert E\rvert-1} - 1$	$+\beta^{\lvert E\rvert-1} - 1$

Tabela 1.7: *Definição dos valores do menor e maior expoentes num sistema de ponto-flutuante.*

Algoritmo 1.3.1 *Estimação de* ε

```
proc macheps(output: ε)
    s ← 1,0
    t ← 2,0
    while (t > 1,0)
        s ← 0,5 * s
        t ← s + 1,0
        if (t ≤ 1,0) then
            ε ← 2,0 * s
        endif
    endwhile
endproc
```

Por exemplo, executando-se esse algoritmo em uma calculadora HP-48SX, teremos como resultado $\varepsilon = 7,2759576141 \times 10^{-12}$. O processador SATURN da HP 48SX utiliza 38 *bits* para representar a mantissa e, portanto, o valor calculado para ε é uma boa aproximação.

1.3.4 Arredondamentos

Conforme salientado anteriormente, qualquer representação de um número real, num computador, será inexata, salvo algumas poucas exceções. Esse erro na representação está associado à base utilizada para representação e ao fato de que, necessariamente, existe um número finito de *bits* para armazenar o número.

Quanto à base, apesar de alguns fabricantes de computadores terem utilizado outras bases (16, no caso do IBM 360 e 8, no BURROUGHS B-6700), em 1960-1970, tipicamente se utiliza a base 2, por ser mais fácil de se implementar os circuitos do processador, usando uma lógica binária. Dessa forma, temos de conviver com o problema de certos números, com representação exata em base decimal, não poderem ser representados de forma exata em base binária.

O outro problema – a limitação no tamanho da palavra para representar um número real – pode ser mitigada ao se aumentar a precisão. Qualquer linguagem de programação científica oferece a possibilidade de se utilizar variáveis em precisão *dupla* e, algumas, em precisão *quádrupla*, i.e., utilizamos duas ou quatro palavras para representar um número real. Outra alternativa é utilizar um processador cuja palavra contenha um maior número de *bits*: o recém-lançado INTEL ITANIUM tem uma palavra de 64 *bits*. Note a diferença – sutil – entre essas duas alternativas: um programa que utilize variáveis em precisão *simples* permitirá se trabalhar com números de diferentes precisões, se utilizarmos dois computadores com palavras de tamanhos diferentes. Por exemplo, um programa em FORTRAN 90 com variáveis de precisão simples – tipo REAL – terá uma precisão de $p = 24$ num computador que utilize o INTEL PENTIUM II, mas, num INTEL PENTIUM 4, o mesmo programa terá uma precisão de $p = 53$.

Assim, com as limitações impostas pela escolha da base e precisão da representação, pode-se perceber que existe um número *finito* de números *representáveis* ou *de máquina*; isso em marcante contraste com os números reais, cuja quantidade é infinita. Mais ainda, entre quaisquer dois números reais, existem infinitos outros números; já em qualquer das representações (ponto-fixo ou ponto-flutuante), se tomarmos dois números representáveis consecutivos, i.e., há uma diferença de 1 no *bit* menos significativo, não há qualquer outro número. Assim, se posicionarmos os números representáveis sobre a reta dos reais, veremos que existem espaços entre cada número representável.

Para demonstrar isso, considere um computador hipotético com uma palavra de 7 bits, e duas representações de números reais:

1. Ponto-fixo: $|s| = 1$, $|e| = 2$, $|d| = 4$;
2. Ponto-flutuante: $|s| = 1$, $|M| = 5$, $|E| = 2$.

As figuras 1.1 e 1.2 mostram a distribuição dos números de máquina ao longo da reta dos reais. Para a representação em ponto-fixo, note que a separação entre os números representáveis é constante: isso é explicado pois o valor de ULP é constante. Já na representação em ponto-flutuante, essa separação aumenta à medida que um número torna-se maior, ou seja, o expoente cresce. Isso pode ser verificado na expressão para ULP, $\varepsilon \times 2^E$, a qual depende do valor do expoente.

Outra característica que pode ser observada é o menor intervalo de representação no sistema de ponto-fixo, bem como uma região de "underflow" muito maior do que no sistema de ponto-flutuante.

Observando as figuras 1.1 e 1.2, podemos verificar a existência de espaços entre os números representáveis. Suponha, então, um número como, por exemplo, $2/3 = 0,66\overline{6}\ldots$, cuja representação em binário é $(0,1010\overline{10}\ldots)_2$. Para fins de explanação consideremos apenas um sistema de ponto-flutuante, apesar das observações a seguir serem válidas para uma representação de ponto-fixo também.

Como 2/3 não tem representação finita em 24 *bits*, temos duas opções para armazená-lo:

1. $(0,101010\ldots101\underline{0})_2$
2. $(0,101010\ldots101\underline{1})_2$

o primeiro número é obtido descartando-se os *bits* $b_{24}b_{25}\ldots$; já o segundo obtém-se descartando-se os *bits* em excesso e somando 1 a b_{24}.

A situação que temos é, portanto, a seguinte: sempre que um número é *não-representável*, devemos escolher outro entre os dois números de máquina, mais próximos daquele. Esses dois números são *arredondamentos* do número original, chamados de *arredondamento por corte* e *por adição*, correspondentes aos itens 1 e 2 acima, respectivamente.

Como estamos aproximando um número não-representável por um outro, o mais próximo dele, nossa representação daquele número traz associada a si um *erro*, o qual pode ser medido de duas formas. Quando um número real x é aproximado por um número $\tilde{x}$, o *erro* é $x - \tilde{x}$. O *erro absoluto* é definido como

$$|x - \tilde{x}| \tag{1.5}$$

e o *erro relativo* é dado por

$$\left|\frac{x - \tilde{x}}{x}\right|. \tag{1.6}$$

Esse último é o mais utilizado por permitir uma comparação mais justa entre quantidades com diferentes relações de magnitude[2].

Vejamos formalmente, agora, quais os *erros* associados a esses arredondamentos. Chamemos de x_c e x_a os números de máquina correspondentes aos arredondamentos por corte e por adição, respectivamente, de um número não-representável x, os quais satisfazem a relação

$$x_c < x < x_a$$

pois

$$\begin{aligned} x_c &= 0,1010\ldots1010| \\ x &= 0,1010\ldots1010|1010\ldots \\ x_a &= 0,1010\ldots1011| \end{aligned}$$

porém, x pode estar mais próximo de x_c ou de x_a, conforme mostrado na figura 1.3.

Escrevendo os números x_c e x_a como

$$x_c = (0,b_0b_1\ldots b_{22}b_{23})_2 \times 2^E \tag{1.7}$$

$$x_a = \left((0,b_0b_1\ldots b_{22}b_{23})_2 + 2^{-24}\right) \times 2^E \tag{1.8}$$

podemos calcular os erros absoluto e relativo associados aos dois arredondamentos como

[2]Por exemplo, um erro de $1m$ na medição da distância entre a Terra e Júpiter é pequeno; porém, um erro de $5cm$ numa incisão num corpo humano pode ser considerado bastante alto.

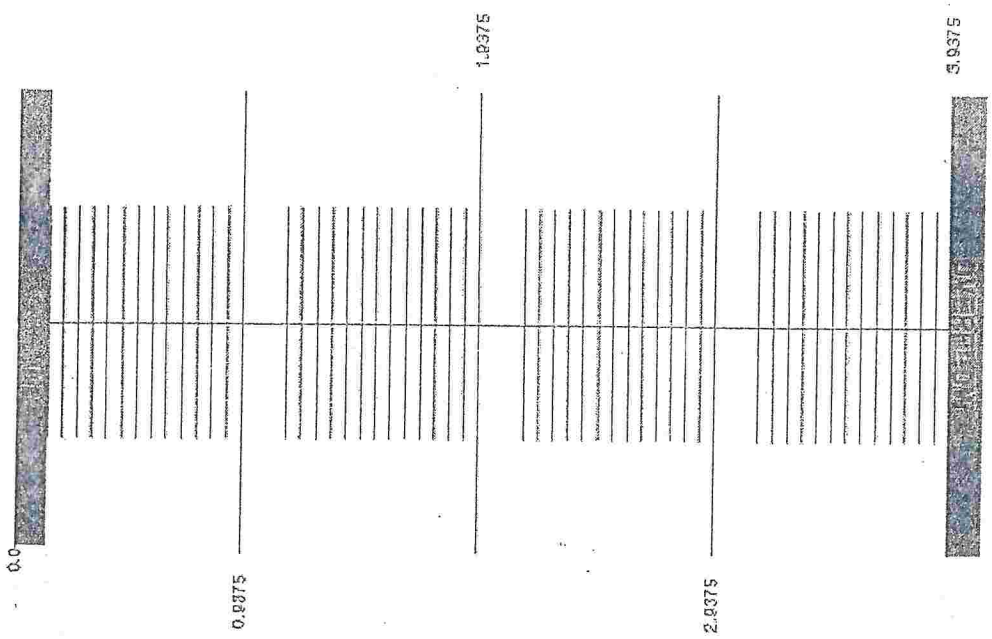

Figura 1.1: *Distribuição de números de máquina em um sistema de ponto-fixo; observe que a distância entre dois números representáveis é a mesma.*

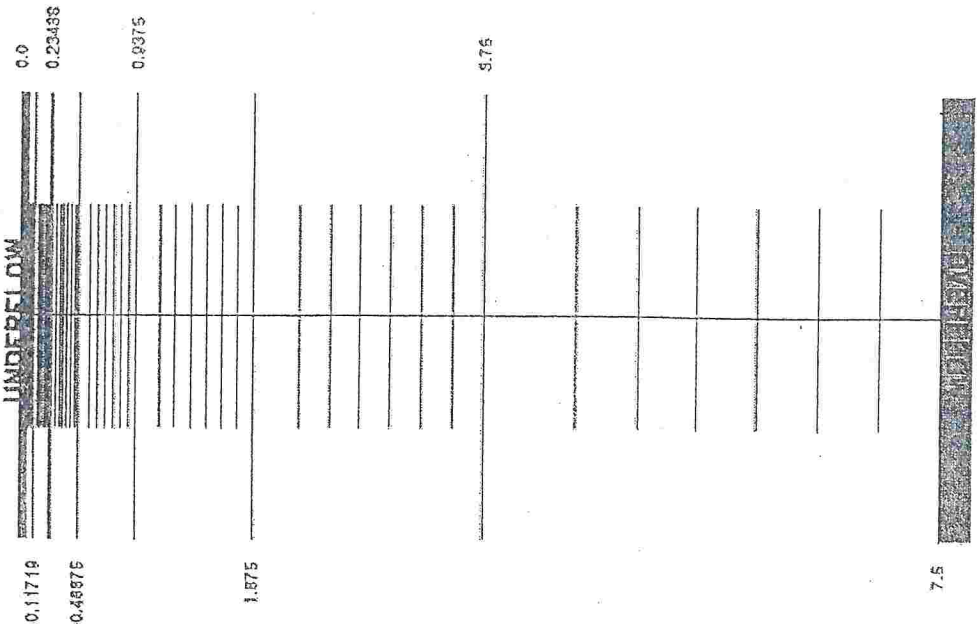

Figura 1.2: *Distribuição de números de máquina em um sistema de ponto-flutuante; aqui, a distância entre dois números representáveis aumenta à medida que se afastam do* 0.

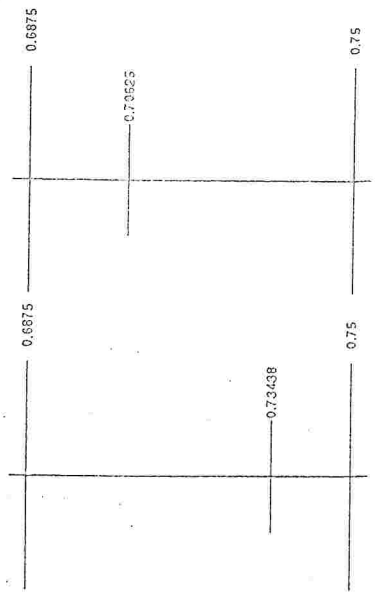

Figura 1.3: *Um número não-representável x pode ser melhor representado por x_c ou x_a. Aqui, consideramos um sistema de ponto-flutuante com quatro dígitos na mantissa, com $x_c = 0,6875 = (0,1011)_2$ e $x_a = 0,75 = (0,1100)_2$; no diagrama à esquerda, $x = 0,70625 = (0,10110100110011\ldots)_2$ e, à direita, $x = 0,734375 = (0,101111)_2$.*

Erros associados ao arredondamento por corte Se o arredondamento por corte foi escolhido, então x encontra-se à esquerda do ponto médio do intervalo $[x_c, x_a]$. Então:

$$|x - x_c| \leq \frac{1}{2}|x_a - x_c| = \frac{1}{2}|(M + 2^{-24}) \times 2^E - M \times 2^E| = \tag{1.9}$$

$$2^{-1} \cdot 2^{-24} \cdot 2^E = 2^{E-25} \tag{1.10}$$

$$\left|\frac{x - x_c}{x}\right| \leq \frac{2^{E-25}}{M \times 2^E} = \frac{2^{-25}}{M} \leq \frac{2^{-25}}{2^{-1}} = 2^{-24} \therefore M \leq \frac{1}{2} \tag{1.11}$$

Erros associados ao arredondamento por adição Se o arredondamento por adição foi escolhido, então x encontra-se à direita do ponto médio do intervalo $[x_c, x_a]$. Por analogia, escrevemos

$$|x - x_a| \leq 2^{E-25} \tag{1.12}$$

$$\left|\frac{x - x_a}{x}\right| \leq 2^{-24} \tag{1.13}$$

Generalizando, podemos dizer que o erro relativo entre x e seu arredondamento x^* é

$$\left|\frac{x - x^*}{x}\right| \leq \varepsilon \tag{1.14}$$

ou

$$x^* = fl\,(x) = x(1 + \delta), \quad |\delta| \leq \varepsilon, \quad \delta = \frac{x^* - x}{x} \tag{1.15}$$

Uma medida muito utilizada para se determinar a qualidade numérica de um valor é o *número de dígitos significativos*, DIGSE. Aplicando logaritmos aos dois lados da expressão 1.14, temos

$$-\log_{10}\left|\frac{x - fl\,(x)}{x}\right| \geq -\log_{10}\varepsilon \tag{1.16}$$

e DIGSE é definido como

$$\text{DIGSE}(x, fl\,(x)) = -\log_{10}\left|\frac{x - fl\,(x)}{x}\right| \tag{1.17}$$

Para uma precisão $p = 24$, $-\log_{10} 2^{-24} \approx 7$, ou seja, temos no mínimo sete casas decimais de precisão.

1.3.5 Operações aritméticas de ponto-flutuante

Na soma e subtração, os expoentes dos dois operandos devem ser iguais. Para tal, seleciona-se o maior dos dois expoentes, e a mantissa e expoente do outro operando são ajustados de tal forma a coincidir os expoentes. Por isso, as operações aritméticas são sempre efetuadas com o dobro de *bits* utilizados para armazenar os números. Uma vez feito o ajuste dos expoentes, basta calcular

$$(a \times r^p) \pm (b \times r^p) = (a \pm b) \times r^p$$

A multiplicação e a divisão são calculadas como

$$\begin{aligned}(a \times r^p) \times (b \times r^q) &= ab \times r^{p+q} \\ (a \times r^p) \div (b \times r^q) &= a \div b \times r^{p-q}\end{aligned}$$

Como essas operações são efetuadas em várias etapas, a cada parcela do processo, deslocam-se os *bits* à esquerda, de forma a sobrar *bits* menos significativos; a cada deslocamento, o expoente deve ser modificado adequadamente.

1.3.5.1 Erros em operações aritméticas de ponto-flutuante

Sempre que dois números de ponto-flutuante sofrerem o efeito de uma das quatro operações aritméticas, as seguintes etapas são efetuadas:

1. A operação é feita de forma "correta", i.e., com o dobro do número de *bits* usados para armazenar cada operando;
2. O resultado é normalizado;
3. É feito o arredondamento, de forma que o resultado normalizado possa ser armazenado na palavra.

O exemplo a seguir mostra por que deve-se efetuar a normalização *antes* do arredondamento.

Exemplo 1.1 *Considere $x = 0,45230 \times 10^{-2}$ e $y = 0,25470 \times 10^{-3}$, em um sistema de ponto-flutuante com cinco casas na mantissa. O resultado de $x \times y$, sem normalização, é*

$$x \times y = 0,0000011520$$

Se efetuarmos o arredondamento, sem normalizá-lo, *o resultado será 0!*

Como mostramos anteriormente, a representação em ponto-flutuante traz associada a si um erro. O exemplo a seguir mostra, no entanto, que uma única operação aritmética simples tem um erro que não excede a ε.

Exemplo 1.2 *Considere $x = 0,31426 \times 10^3$ e $y = 0,92577 \times 10^5$ e, calculando as quatro operações aritméticas, temos:*

$$\begin{aligned}x + y &= 0,9289100000 \times 10^5 \\ x - y &= -0,9226274000 \times 10^5 \\ x \times y &= 0,2909324802 \times 10^8 \\ x \div y &= 0,3394579647 \times 10^{-2}\end{aligned}$$

e, arredondando para cinco casas decimais, por adição, temos

		erro relativo
fl $(x+y)$	$0,92891 \times 10^{5}$	$8,5 \times 10^{-6} < 10^{-5}$
fl $(x-y)$	$-0,92263 \times 10^{5}$	$2,3 \times 10^{-6} < 10^{-5}$
fl $(x \times y)$	$0,29093 \times 10^{8}$	$2,8 \times 10^{-6} < 10^{-5}$
fl $(x \div y)$	$0,33946 \times 10^{-2}$	$6,0 \times 10^{-6} < 10^{-5}$

onde 10^{-5} *é o* ε *dessa representação em cinco casas decimais.*

De forma genérica, podemos dizer que, se x e y são números representáveis, então para uma operação aritmética qualquer $\odot$,

$$fl\,(x \odot y) = (x \odot y)(1+\delta), \quad |\,\delta\,| \leq \varepsilon \tag{1.18}$$

e, se x e y não são números representáveis, então

$$fl\,(fl\,(x) \odot fl\,(y)) = (x(1+\delta_1) \odot y(1+\delta_2))\,(1+\delta_3), \quad |\,\delta_{1,2,3}\,| \leq \varepsilon \tag{1.19}$$

A equação (1.19) nos diz que o erro associado ao encadeamento de operações aritméticas pode ser maior do que ε. Considere o exemplo abaixo:

$$\begin{aligned} fl\,(x(y+z)) &= (x(fl\,(y+z))(1+\delta_1), \quad |\,\delta_1\,| \leq 2^{-24} \\ &= (x(y+z)(1+\delta_2))(1+\delta_2), \quad |\,\delta_2\,| \leq 2^{-24} \\ &= x(y+z)(1+\delta_1+\delta_2+\delta_1\delta_2) \\ &\approx x(y+z)(1+\delta_1+\delta_2) \therefore |\,\delta_1+\delta_2\,| \leq 2^{-23} \\ &\approx x(y+z)(1+\delta_3), \quad |\,\delta_3\,| \leq 2^{-23}, \delta_1\delta_2 \ll \delta_3 \end{aligned}$$

Esse exemplo nos leva a supor que, caso a quantidade de operações aritméticas a serem feitas seja muito grande, então o erro crescerá proporcionalmente. O teorema a seguir mostra que essa hipótese é verdadeira.

Teorema 1.3.1 *Sejam* $x_0, x_1, \ldots, x_n$ *números representáveis positivos, e* ε *a unidade de arredondamento da máquina. Então, o erro relativo de arredondamento ao se calcular* $\sum_{i=0}^{n} x_i$ *na ordem natural, i.e.* $x_0 + x_1 + \ldots + x_n$*, é de no máximo* $(1+\varepsilon)^n - 1$ *ou, aproximadamente,* $n\varepsilon$*.*
Prova: Seja $S_k = x_0 + x_1 + \ldots + x_k$ *e* fl (S_k)*, as quais podem ser representadas pelas fórmulas de recorrência*

$$\begin{cases} S_0 &= x_0 \\ S_{k+1} &= S_k + x_{k+1}, \quad k \geq 0 \end{cases} \tag{1.20}$$

$$\begin{cases} \text{fl}\,(S_0) &= x_0 \\ \text{fl}\,(S_{k+1}) &= \text{fl}\,(\text{fl}\,(S_k) + x_{k+1}), \quad k \geq 0 \end{cases} \tag{1.21}$$

e chamemos de ρ_k *e* δ_k *aos erros relativos associados a* fl (S_k) *e* fl (S_{k+1})*,*

$$\rho_k = \frac{\text{fl}\,(S_k) - S_k}{S_k} \tag{1.22}$$

$$\delta_k = \frac{\text{fl}\,(S_{k+1}) - (\text{fl}\,(S_k) + x_{k+1})}{\text{fl}\,(S_k) + x_{k+1}} \tag{1.23}$$

de onde

$$\begin{aligned} \rho_{k+1} &= \frac{\text{fl}\,(S_{k+1}) - S_{k+1}}{S_{k+1}} \\ &= \frac{(\text{fl}\,(S_k) + x_{k+1})(1+\delta_k) - (S_k + x_{k+1})}{S_{k+1}} \therefore \text{fl}\,(S_k) = S_k\rho_k + S_k \textit{(por (1.22))} \therefore \\ &= \frac{(S_k(1+\rho_k) + x_{k+1})(1+\delta_k) - (S_k + x_{k+1})}{S_{k+1}} \end{aligned}$$

e, rearranjando os termos, obtemos

$$\rho_{k+1} = \delta_k + \rho_k \frac{S_k}{S_{k+1}} \tag{1.24}$$

Como, por definição, $S_k < S_{k+1}$ *e* $|\delta_k| \leq \varepsilon$,

$$|\rho_{k+1}| \leq \varepsilon + |\rho_k|(1+\varepsilon) = \varepsilon + \theta|\rho_k|, \quad \theta = 1+\varepsilon$$

podemos então escrever:

$$\begin{aligned} |\rho_0| &= 0 \\ |\rho_1| &\leq \varepsilon \\ |\rho_2| &\leq \varepsilon + \theta\varepsilon \\ |\rho_3| &\leq \varepsilon + \theta(\varepsilon + \theta\varepsilon) = \varepsilon + \theta\varepsilon + \theta^2\varepsilon \\ &\vdots \end{aligned}$$

ou

$$\begin{aligned} |\rho_n| &\leq \varepsilon + \theta(\varepsilon + \theta\varepsilon) = \varepsilon + \theta\varepsilon + \theta^2\varepsilon + \ldots + \theta^{n-1}\varepsilon = \\ &= \varepsilon(1 + \theta + \ldots + \theta^{n-1}) = \\ &= \varepsilon\frac{\theta^n - 1}{\theta - 1} = \\ &= \varepsilon\frac{(1+\varepsilon)^n - 1}{\varepsilon} = \\ &= (1+\varepsilon)^n - 1 \end{aligned}$$

e, pelo binômio de Newton, tem-se

$$(1+\varepsilon)^n - 1 = 1 + \binom{n}{1}\varepsilon = \binom{n}{2}\varepsilon^2 + \ldots - 1 \approx n\varepsilon. \diamond$$

1.4 Perda de dígitos significativos

Apesar de erros de arredondamento serem inevitáveis e difíceis de controlar, existem alguns tipos de erros, em computações numéricas, que podem ser evitados.

Por exemplo, suponha a subtração de dois números x e y próximos entre si:

$$\begin{aligned} x &= .3721478693 \\ y &= .3720230572 \\ x - y &= .0001248121 \end{aligned}$$

Se o computador utilizado oferecer apenas cinco dígitos decimais na mantissa, teríamos:

$$\begin{aligned} fl\,(x) &= .37215 \\ fl\,(y) &= .37202 \\ fl\,(x) - fl\,(y) &= .00013 \end{aligned}$$

Nesse caso, o erro relativo é bastante grande, da ordem de 4%.

1.4.1 Subtração de valores quase idênticos

Como regra, devemos evitar situações que levem à subtração de valores quase idênticos - normalmente causados por expressões inadequadas do ponto de vista numérico.

Exemplo 1.3 *Considere a expressão*

$$y \leftarrow \sqrt{x^2+1} - 1 \tag{1.25}$$

Ora, para $x < 1$, *teremos* $x^2 \ll 1$ *e, portanto,* $\sqrt{x^2+1} \approx 1$. *No entanto, se reescrevermos a expressão acima como*

$$\widetilde{y} \leftarrow \left(\sqrt{x^2+1} - 1\right)\left(\frac{\sqrt{x^2+1}+1}{\sqrt{x^2+1}+1}\right) = \frac{x^2}{\sqrt{x^2+1}+1} \tag{1.26}$$

eliminaremos esse problema. Por exemplo, em uma calculadora HP-48SX, se $x = 10^{-6}$, *teremos:*

$$\begin{aligned} y &\leftarrow \sqrt{0,000001^2+1} - 1 = \sqrt{10^{-12}+1} - 1 = 1 - 1 = 0 \\ \widetilde{y} &\leftarrow \frac{0,000001^2}{\sqrt{0,000001^2+1}+1} = \frac{10^{-12}}{\sqrt{1}+1} = \frac{10^{-12}}{2} = 5 \times 10^{-13} \end{aligned}$$

Por que, no exemplo acima, $10^{-12} + 1 = 1$? *Ocorre que, se um dos operandos for menor do que* ε, *então ele será desconsiderado (pois* ε *é o menor número representável tal que* $1 + \varepsilon \neq 1$*). Como* $(10^{-6})^2 < \varepsilon$ *(*$= 7,2759576141 \times 10^{-12}$ *na HP-48SX), houve o cancelamento catastrófico na subtração.*

Exemplo 1.4 *Considere as expressões para as duas raízes de uma equação de segundo grau,*

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{1.27}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{1.28}$$

Se $b^2 \gg 4ac$, *então a expressão* $b^2 - 4ac$ *envolve cancelamento e* $\sqrt{b^2 - 4ac} \approx |b|$, *de tal forma que as expressões sofrerão cancelamento catastrófico, dependendo do sinal de* b.

Nesse caso, se multiplicarmos as equações acima por expressões do tipo

$$\frac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}}, \quad \frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}$$

poderemos calcular as raízes como segue:

1. *Se* $b^2 \gg 4ac$ *e* $b > 0$ *então* x_2 *é calculado por (1.28) e*

$$x_1 = \frac{2c}{-b - \sqrt{b^2 - 4ac}} = \frac{c}{ax_2} \tag{1.29}$$

2. *Se* $b^2 \gg 4ac$ *e* $b < 0$ *então* x_1 *é calculado por (1.27) e*

$$x_2 = \frac{2c}{-b + \sqrt{b^2 - 4ac}} = \frac{c}{ax_1} \tag{1.30}$$

Por exemplo, seja a equação $x^2 - 10^6 x + 1 = 0$. *Calculando* x_1 *e* x_2 *através das expressões usuais, utilizando o* software MAPLE *com precisão de 10 dígitos decimais, em um computador* PENTIUM II, *temos:*

$$\begin{aligned} x_1 &= \frac{10^6 + \sqrt{10^{12} - 4}}{2} = \frac{10^6 + 10^6}{2} = 10^6 \\ x_2 &= \frac{10^6 - \sqrt{10^{12} - 4}}{2} = \frac{10^6 - 10^6}{2} = 0 \end{aligned}$$

A raiz x_2 foi calculada de forma errada, pois sofreu cancelamento catastrófico. No entanto, se recalculamo-la usando (1.30), temos

$$x_2 = \frac{2}{10^6 + \sqrt{10^{12} - 4}} = \frac{2}{10^6 + 10^6} = 10^{-6}$$

Ao substituirmos esse valor na equação, teremos $10^{-12} - 1 + 1$, o qual pode ser considerado como aproximando 0 para a precisão utilizada. Note que, para esse exemplo particular, x_1 não é raiz da equação, pois $10^{12} - 10^6 10^6 + 1 \neq 0$.

1.4.2 Teorema sobre a perda de precisão

Uma questão que surge a partir dos exemplos anteriores é a seguinte: *Quantos dígitos binários significativos são perdidos na subtração $x - y$ quando $x \approx y$?* O teorema a seguir nos dá limitantes extremos para o número de dígitos binários perdidos nessa situação, baseado na relação $|1 - y/x|$, o que nos dá uma medida de quão próximo x é de y.

Teorema 1.4.1 *Se x e y são números em ponto-flutuante binários positivos, normalizados, tal que $x > y$ e*

$$2^{-q} \leq 1 - \frac{y}{x} \leq 2^{-p}$$

então no máximo q e no mínimo p dígitos binários significativos são perdidos na subtração $x - y$.
Prova. Considerando apenas o extremo superior da desigualdade, temos que x e y são da forma

$$\begin{aligned} x &= r \times 2^n, \quad \left(\frac{1}{2} \leq r < 1\right) \\ y &= s \times 2^m, \quad \left(\frac{1}{2} \leq s < 1\right) \end{aligned}$$

Como $x > y$, por hipótese, o expoente de y deverá ser igualado ao de x antes de se realizar a subtração (note que, como não pode haver dígitos à esquerda do ponto binário, sempre se faz com que o menor número iguale seu expoente ao maior, introduzindo zeros imediatamente à direita do ponto binário). Logo, y deve ser escrito como

$$y = (s \times 2^{m-n}) \times 2^n$$

e, daí,

$$x - y = (r - s \times 2^{m-n}) \times 2^n$$

A mantissa desse número satisfaz a seguinte relação:

$$r - s \times 2^{m-n} = r\left(1 - \frac{s \times 2^m}{r \times 2^n}\right) = r\left(1 - \frac{y}{x}\right) < 2^{-p}$$

Para normalizar a representação de $x - y$, um deslocamento de ao menos p bits para a esquerda é necessário. Então, ao menos p zeros são inseridos ao final da mantissa, efetivamente perdendo p bits de precisão. ◇

O exemplo a seguir ilustra a utilização desse resultado.

Exemplo 1.5 *Suponha uma mantissa de 5 dígitos decimais e que $x = 0,31457 \times 10^5$ e $y = 0,31453 \times 10^4$ e que os cálculos sejam efetuados com o dobro de dígitos. Ora, para calcular $x - y$, temos:*

$$\begin{aligned} x &= 0,31457\ 00000 \times 10^5 \\ y &= 0,03145\ 30000 \times 10^5 \\ x - y &= 0,28311\ 70000 \times 10^5 \\ \text{fl}\,(x - y) &= 0,28311 \times 10^5 \end{aligned}$$

como foi necessário inserir um dígito "0" após o ponto decimal de y, espera-se que se perderá um dígito ao fazer a normalização, como pode-se verificar pelo resultado. O erro relativo, no caso, é

$$\left| \frac{0,283117 \times 10^5 - 0,28311 \times 10^5}{0,283117 \times 10^5} \right| = 2,4724760435 \times 10^{-5} \not\leq 10^{-5}$$

o que demonstra que, efetivamente, houve perda de dígitos significativos na subtração.

Exemplo 1.6 *Considere a expressão* $y = x - \text{sen}x$. *Como* $\text{sen}x \approx x$ *para* $x \ll 1$, *ocorrerá perda de dígitos significativos em y. Proponha uma forma alternativa para calcular y e estipule um intervalo para x no qual pode-se utilizar a expressão original.*
Solução: Usando a série de Taylor para sen$x$, *temos:*

$$\begin{aligned} y &= x - \text{sen}x \\ &= x - \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \right) \\ &= \left(\frac{x^3}{3!} - \frac{x^5}{5!} + \frac{x^7}{7!} - \ldots \right) \end{aligned}$$

e, para $x \approx 0$, *podemos truncar a série como segue, utilizando apenas quatro termos:*

$$\widetilde{y} = (x^3/6)(1 - (x^2/20)(1 - (x^2/42)(1 - x^2/72)))$$

Usando o Teorema da Perda de Precisão, podemos exigir que apenas um *bit seja perdido se escrevermos*

$$1 - \frac{\text{sen}x}{x} \geq \frac{1}{2}, \quad x > 0,$$

Essa desigualdade é satisfeita se $|x| \geq 1,9$ *e, nessa situação, podemos usar a expressão original. Para* $0 < x < 1,9$, *devemos usar a expressão baseada na série truncada de Taylor, pois ela elimina o problema.*

Suponha $x = 10^{-6}$ *em uma calculadora HP-48SX. Então, teremos:*

$$\begin{aligned} y &= 10^{-6} - \text{sen}(10^{-6}) = 0,000001 - 0,000001 = 0 \\ \widetilde{y} &= 1,66666666667 \times 10^{-19} \end{aligned}$$

1.5 Condicionamento de um problema

O condicionamento de um problema diz respeito a quão exato podemos resolvê-lo em uma dada precisão de ponto-flutuante, *independentemente* do algoritmo utilizado para resolvê-lo.

Seguindo a derivação em [12], e supondo que desejamos avaliar uma função $y = f(x)$, já sabemos que qualquer operação de ponto-flutuante acarretará a existência de um erro. Logo, o que efetivamente calcula-se é uma aproximação

$$fl\,(y) = fl\,(f(fl\,(x)))$$

mas, por simplicidade, assumimos que $fl\,(f) = f$. Podemos calcular, então, o erro relativo em y como

$$\frac{fl\,(y) - y}{y} = \frac{f(fl\,(x)) - f(x)}{fl\,(x) - x} \times \frac{x}{f(x)} \times \frac{fl\,(x) - x}{x}.$$

O termo $\frac{f(fl\,(x)) - f(x)}{fl\,(x) - x}$ é uma aproximação para $f'(x)$. Assim,

$$\frac{fl\,(y) - y}{y} \approx \kappa_f(x) \times \frac{fl\,(x) - x}{x} \tag{1.31}$$

onde

$$\kappa_f(x) = \frac{|x| \times |f'(x)|}{|f(x)|} \tag{1.32}$$

a qual é chamada de *número de condição de f em x*. Esse fator é que mede o quanto os erros de arredondamento em x são amplificados ao se avaliar $f(x)$.

Então, para se avaliar o número de dígitos corretos em $fl\,(y)$, aplicamos logaritmos aos dois membros da equação (1.31),

$$-\log_{10}\left(\frac{fl\,(y)-y}{y}\right) \approx -\log_{10}\left(\frac{fl\,(x)-x}{x}\right) - \log_{10}\kappa_f(x)$$

Note que $-\log_{10}\left(\frac{fl\,(x)-x}{x}\right)$ é aproximadamente igual a 7 para $p=24$ e a 16 para $p=53$.

1.6 Computações estáveis e instáveis

Um processo numérico é dito *instável* se pequenos erros ocorridos num passo são ampliados nos passos seguintes, degradando a exatidão do processo.

Considere, por exemplo, a seqüência de números dada por

$$\begin{cases} x_0 = 1 \\ x_1 = \frac{1}{3} \\ x_{n+1} = \frac{13}{3}x_n - \frac{4}{3}x_{n-1}, \quad n \geq 1 \end{cases}$$

a qual gera os números

$$x_n = \left(\frac{1}{3}\right)^n$$

pois $x_0 = \frac{1}{3}^0 = 1$, $x_1 = \frac{1}{3}^1 = \frac{1}{3}$. Para $n = m+1$, temos:

$$\begin{aligned} x_{m+1} &= \frac{13}{3}x_m - \frac{4}{3}x_{m-1} = \frac{13}{3}\left(\frac{1}{3}\right)^m - \frac{4}{3}\left(\frac{1}{3}\right)^{m-1} \\ &= \left(\frac{1}{3}\right)^{m-1}\left[\frac{13}{3} - \frac{4}{3}\right] = \left(\frac{1}{3}\right)^{m+1} \end{aligned}$$

Utilizando a forma de recorrência acima para gerar os números, teremos:

n	x_n	$(1/3)^n$	erro relativo
0	1,00000000000000000000	1,00000000000000000000	0,00000000000000000000
1	0,33333333333333310000	0,33333333333333310000	0,00000000000000000000
2	0,11111111111111040000	0,11111111111111111000	0,00000000000000001498801
3	0,03703703703703628000	0,03703703703703728000	0,00000000000000270795865
4	0,01234567901234251400	0,01234567901234567700	0,00000000000000256154473
5	0,00411522633743584400	0,00411522633744855830	0,00000000000003079755201
6	0,00137174211243214560	0,00137174211248285290	0,00000000000369655985511
7	0,00045724737062476524	0,00045724737082761756	0,00000000044359429606
8	0,00015241578946454185	0,00015241579027587250	0,00000000532314044454
9	0,00005080526017996764	0,00005080526342529083	0,00000006387769640489
10	0,00001693507482713733	0,00001693508780843027	0,00000076532360861972
11	0,00000564497734430494	0,00000564502926947675	0,00000919838841059637
12	0,00000188146872247166	0,00000188167642315892	0,00011038066093717230
13	0,00000062639467163726	0,00000062722547438630	0,01324567931256533500
14	0,00000020575194713260	0,00000020907515812876	0,01589481517508941800
15	0,00000005639887539161	0,00000006969171937625	0,19073778210108555000
16	-0,00000002994080281313	0,00000002323057312541	2,28885338521303970000
17	-0,00000020494197937907	0,00000000774352437514	27,46624062556484000000

Os valores acima demonstram que o algoritmo é instável, e qualquer erro presente em x_n é multiplicado por 13/3 em x_{n+1}. Portanto, há a possibilidade de que o erro existente em x_3 (da ordem de 10^{-15}) seja propagado para x_{17} por um fator $(13/3)^{14} \approx 10^9$; ou seja, o erro em x_{17} devido unicamente a x_3 pode ser de 10^{-4}, que não é desprezível. Além disso, os erros devido aos demais números $x_4, x_5, \ldots, x_k$ são propagados para x_{17} por fatores da forma $(13/3)^k$.

1.7 Desastres causados por erros aritméticos no computador

Esse capítulo apresentou os conceitos ligados à computação numérica; dentre estes, certamente o conceito de *erro* associado a *qualquer* cálculo numérico é o mais importante. Como não há como evitá-los, é necessário que o programador e/ou analista numérico saiba como tratá-los de forma que a ocorrência deles não leve a falhas catastróficas. Infelizmente, isso nem sempre é levado em conta, no dia-a-dia, e desastres ocorrem, como os dois que citamos a seguir.

1.7.1 Falha do sistema de mísseis "Patriot"

Durante a Guerra do Golfo, em 1991, o Iraque lançou inúmeros mísseis terra-terra "Scud" (de fabricação soviética) contra Israel e Arábia Saudita. A fim de se protegerem contra esses ataques, as tropas norte-americanas instalaram baterias de mísseis terra-ar "Patriot", os quais haviam sido projetados no início da década de 70 para destruírem mísseis cruzeiro e aeronaves soviéticas (voando a uma velocidade média de 2Mach), numa eventual guerra entre a OTAN e o Pacto de Varsóvia.

Uma bateria de mísseis "Patriot" consiste de uma unidade de controle computadorizada; de um radar de detecção; e de até 6 lançadores quádruplos de mísseis. A unidade de controle dispõe de um relógio que marca o tempo em décimos de segundo, armazenados em uma palavra inteira de 24 bits; os cálculos de determinação das janelas de confirmação e de engajamento (regiões no céu dentro do qual o possível alvo deve ser detectado pelo radar para que possa os mísseis "Patriot" sejam lançados) são feitos em ponto fixo, também com 24 bits.

No dia 25 de fevereiro de 1991, uma bateria "Patriot" instalada em Dharan, na Arábia Saudita, deixou de interceptar um míssil "Scud" que se aproximava. Como resultado, 28 soldados norte-americanos foram mortos devido à explosão do míssil "Scud".

Os resultados da investigação, de acordo com [11], indicaram que os mísseis "Patriot" não engajaram o "Scud" (apesar dos radares haverem detectado o míssil iraquiano) devido a um erro numérico de arredondamento. O tempo medido pelo relógio da unidade de controle é multiplicado por 1/10 para representar o tempo em segundos, e armazenado em 23 bits, no formato de ponto fixo; ocorre que 1/10 é um número que não tem representação finita em binário:

$$(1/10)_{10} = (0,0001\ 1001\ 1001\ 1001\ 1001\ 1001\ 1001\ 100\ldots)_2$$

Como a palavra usada para armazenar o relógio é de 24 bits, o erro é de aproximadamente

$$(0,0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 100\ldots)_2 \approx (0,0000\ 0009\ 5)_{10}.$$

É óbvio que, à medida que o tempo de operação da bateria de mísseis aumenta, maior será o erro no tempo calculado. Como esse tempo é usado para se calcular as janelas de detecção e engajamento de um alvo, isso irá causar um deslocamento da janela para baixo (i.e., a altitude na qual se espera que o alvo aparecerá na próxima varredura do radar da bateria será *menor* do que a que ele se encontra). Com isso, o míssil continuará trafegando em direção ao seu alvo, porém não será detectado e, portanto, os mísseis "Patriot" não serão disparados.

Ocorre que, devido às características do sistema "Patriot", a fim de se maximizar as chances de derrubada do míssil atacante, este deve encontrar-se no "meio" da janela de engajamento, a qual tem um comprimento de $274m$. No dia 11 de fevereiro de 1991 (duas semanas antes da falha do sistema), verificou-se que após $8h$ de operação contínua, a janela sofria um deslocamento de $55m$; por extrapolação, após $20h$ de uso contínuo, esse deslocamento seria de $137m$, e a partir de então, não seria mais possível detectar um míssil atacante.

Quando ocorreu a falha, a bateria que deveria ter engajado o "Scud" estava operando continuamente por $100h$! Dessa forma, o erro acumulado era de

$$0,000000095 \times 100 \times 3600 \times 10 = 0,34s.$$

Um míssil "Scud" viaja a uma velocidade terminal de $1.676m/s$; em $0,34s$, ele percorre a distância de $569,84m$. O deslocamento da janela de detecção, após $100h$ de operação, era de $687m$. Logo,

não havia como o "Scud" ser detectado, já que a janela encontrava-se a uma altitude inferior à dele.

1.7.2 Explosão do foguete Ariane 5

No dia 4 de junho de 1996, o primeiro foguete Ariane 5, construído pela Agência Espacial Européia, foi destruído pelo sistema de controle de falha apenas 40*s* após o lançamento da sua base em Khourou, na Guiana Francesa. O Ariane 5 havia sido desenvolvido após 10 anos de trabalho, a um custo de 7 bilhões de dólares. O custo do foguete, bem como da carga útil transportada, era de 500 milhões de dólares.

Os resultados da investigação, após duas semanas do incidente, indicaram que o problema encontrava-se no "software" de guiagem inercial. Um número em ponto-flutuante, armazenado numa palavra de 64 bits, e que representava a velocidade horizontal em relação à plataforma de lançamento, foi convertido para um número inteiro, no formato sinal-e-magnitude, de 16 bits. Como a velocidade era superior a 32.768 (o maior número representável em 15 bits), ocorreu uma falha na conversão e o programa deixou de funcionar.

1.8 Exercícios

Exercício 1.1 *Considere um sistema de ponto-flutuante,* $F = (2, 24, 8)$*, no qual os números apresentam uma mantissa tal que* $1 \leq M < 2$*. Determine os valores caracterizadores desse sistema.*

Exercício 1.2 *Suponha que alguém recebeu a tarefa de projetar o sistema de ponto-flutuante de um computador com uma palavra de* 16 bits*, para uma aplicação envolvendo a medição de temperaturas próximas a zero. Qual o tamanho dos campos* M *e* E *que seria mais indicado? Justifique a sua resposta.*

Exercício 1.3 *Mostre que, normalmente,* $\text{fl}\,[\text{fl}\,(xy)z] \neq \text{fl}\,[x\text{fl}\,(yz)]$*; escreva um exemplo e um contra-exemplo.*

Exercício 1.4 *Se no máximo 2 bits de precisão podem ser perdidos ao se calcular* $y = \sqrt{x^2+1} - 1$*, qual a restrição que deve ser imposta a* x*?*

Exercício 1.5 *Calcule os valores de* x *para os quais* $f(x) = \text{sen}(x)$ *pode ter um grande número de condição.*

Capítulo 2

Cálculo de Raízes de Funções Não-Lineares

2.1 Introdução

Consideramos aqui o problema de se determinar um valor real para o qual uma determinada função $f(x)$ se anula. Seja então

$$\begin{array}{rccc} f: & \mathbb{R} & \rightarrow & \mathbb{R} \\ & x & \mapsto & f(x) \end{array}$$

e deseja-se determinar um valor $\xi \in \mathbb{R}$ tal que $|f(\xi)| \approx 0$. O processo de busca de ξ é feito basicamente de duas formas. Na primeira, determina-se uma seqüência de intervalos que contém ξ; na segunda, dada uma estimativa inicial x_0 para ξ, refina-se sucessivamente essa estimativa através de alguma fórmula, até que se obtenha uma boa aproximação para a raiz desejada. Existem inúmeros métodos para se determinar tais raízes, baseados nessas duas formas de busca.

Note que esses métodos não determinam *todas* as raízes de $f(x) = 0$. Para ter uma idéia mais precisa da localização das raízes de uma equação é preciso encontrar uma seqüência de subintervalos distintos, tais que cada subintervalo contivesse exatamente uma raiz real e cada raiz real estivesse contida em um subintervalo. Este processo é chamado de *separação das raízes de uma função.*

Teorema 2.1.1 *Teorema de Bolzano: Seja f uma função contínua em um intervalo $[a,b]$. Então,*

1. *Se $f(a)f(b) < 0$, então existe um número ímpar $(1,3,5,\ldots)$, de raízes reais em $[a,b]$;*
2. *Se $f(a)f(b) > 0$, pode existir um número par $(0,2,4,\ldots)$ de raízes reais em $[a,b]$.*
3. *Supondo que f e sua derivada f' sejam contínuas em $[a,b]$ e que o sinal de f' seja constante neste intervalo tem-se:*
   (a) *se $f(a)f(b) < 0$, então existe uma única raiz real em $[a,b]$;*
   (b) *se $f(a)f(b) > 0$, então não existe raiz real em $[a,b]$.*

Exemplo 2.1 *Determine graficamente os intervalos que contém cada uma das raízes reais das funções $f(x) = x^3 - \text{sen}(x)$ e $p(z) = z^3 - 4z^2 + 4z - 1$.*

Além disso, é possível que existam raízes de multiplicidade maior do que um; nesse caso, ainda que um método consiga determinar uma delas, a determinação da multiplicidade dela deve ser feita posteriormente. Outro problema que afeta a determinação numérica de uma raiz é quando existem duas raízes tão próximas numericamente que a não se pode garantir para qual delas um determinado método irá convergir.

Nesse capítulo, apresentaremos os métodos da bissecção, da posição falsa, de Newton e da secante.

2.2 Método da Bissecção

O método da bissecção é um dos mais métodos mais simples para se obter uma raíz de uma função. Ele baseia-se na divisão sucessiva, ao meio, de um intervalo $[a, b]$, no qual é garantido que há uma raiz da equação $f(x) = 0$, i.e., $sign(f(a)) \neq sign(f(b))$ - daí o seu nome.

Formalmente, se f é uma função contínua no intervalo $[a, b]$, e se $f(a)f(b) < 0$, então f tem um zero em (a, b). Como há troca de sinal da função f avaliada nos extremos do intervalo, em algum ponto $a < x < b$ a curva da função f cruzou o eixo das abscissas e, portanto, *existe ao menos uma raiz* em (a, b).

Considere, agora, a figura 2.1.

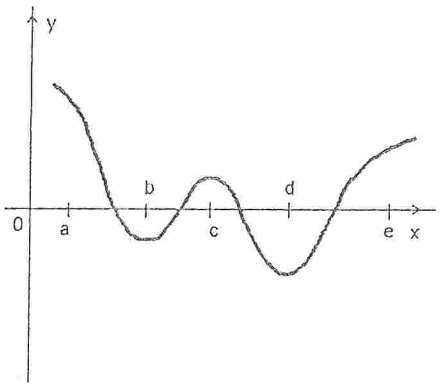

Figura 2.1: O método da bissecção escolhe o intervalo esquerdo.

Veja que os intervalos $[a, b]$, $[b, c]$, $[c, d]$ e $[d, e]$ contém as quatro raízes mostradas, e que $f(a)f(b) < 0$; $f(b)f(c) < 0$; $f(c)f(d) < 0$; e $f(d)f(e) < 0$. No entanto, *se* tivéssemos considerado o intervalo $[a, e]$, teríamos $f(a)f(e) > 0$, e poderíamos concluir que não existem raízes neste intervalo, o que é obviamente errado. Com isso, mostramos que a condição $f(a)f(b) < 0$ é necessária, porém não suficiente. Devemos, portanto, selecionar intervalos que contenham raízes, geralmente com o auxílio de um gráfico da função.

O método da bisseção funciona da seguinte maneira: dados os extremos do intervalo, a e b $(a < b)$, calcula-se um ponto c como o ponto médio daquele intervalo, $c = \frac{a+b}{2}$. Verifica-se, então, se $f(a)f(c) < 0$; se a desigualdade for satisfeita, existe um zero no intervalo (a, c). Caso contrário, verificamos se $f(c)f(b) < 0$ e, em sendo verdade, temos um zero (no mínimo) entre (c, b).

Se, em um dado intervalo $[a, b]$, com $f(a)f(b) < 0$, existir mais de uma raiz, não é possível determinar, de antemão, qual raiz o método da bissecção localizará, conforme pode ser visto nas figuras 2.2 e 2.3.

Obviamente, se $f(c) = 0$, teremos obtido um dos zeros no intervalo $[a, b]$. No entanto, conforme vimos anteriormente, devido a erros de arredondamento, dificilmente uma quantidade é *exatamente* igual a zero; devemos, então, testar se $|f(c)| < \epsilon$, onde ϵ é uma tolerância previamente especificada.

Existem, no entanto, situações em que, devido à natureza da função, aquela condição não é satisfeita. Considere a figura 2.4-(a); veja que $f(a)f(b) < 0$ e que $f(0) = 0$. Porém, como os dois ramos da curva são assíntotas ao eixo das ordenadas, os intervalos gerados poderão ser tais que $f(c) > 0$, mas $|b_n - a_n|$ tende a zero. Portanto, é interessante considerarmos um outro critério

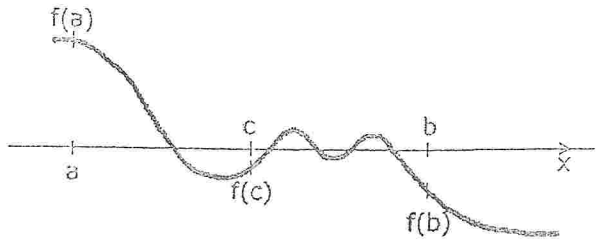

Figura 2.2: O método da bissecção escolhe o intervalo esquerdo.

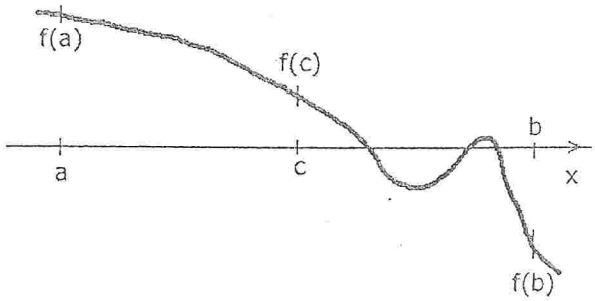

Figura 2.3: O método da bissecção escolhe o intervalo direito.

de parada: se $|b_n - a_n| < \delta$ (onde δ é, também, previamente especificado), então o processo de aproximação é interrompido. Já na figura 2.4-(b), temos a situação contrária.

Um algoritmo para o método da bissecção pode ser escrito como segue:

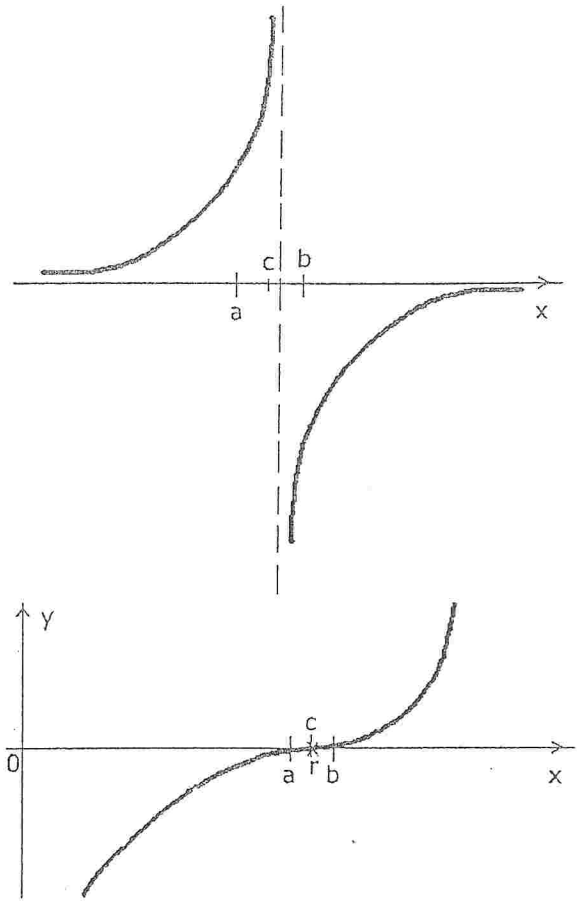

Figura 2.4: (a) $|f(c)| > \epsilon$ e $|b_n - a_n| < \delta$; (b) $|f(c)| < \epsilon$ e $|b_n - a_n| > \delta$.

Algoritmo 2.2.1 *Método da bissecção*

```
proc bissecção(input: a, b, k_max, δ, ε; output: c)
    u ← f(a)
    v ← f(b)
    e ← b − a
    if (sign(u) = sign(v)) then
      "não pode proceder"
    else
     k ← 1
     w ← 1
     while ((k ≤ k_max) AND (|e| ≥ δ) AND (|w| ≥ ε))
          e ← e/2
          c ← a + e
          w ← f(c)
          if (sign(w) ≠ sign(u)) then
            b ← c
            v ← w
          else
            a ← c
            u ← w
          endif
          k ← k + 1
     endwhile
    endif
endproc
```

2.3 Método da posição falsa

O método da posição falsa é uma modificação do método da bissecção. Conforme visto anteriormente, é possível que o ponto médio do intervalo de busca não seja o mais próximo da raiz contida naquele intervalo.

O método da posição falsa faz um refinamento sucessivo de um intervalo $[a, b]$ utilizando a intersecção da reta secante (ou corda) a $f(x)$ – a qual passa pelos pontos $(a, f(a))$ e $(b, f(b))$ – com o eixo dos x. Esse ponto de intersecção será um dos extremos do novo intervalo de busca, ou será uma boa aproximação para a raiz.

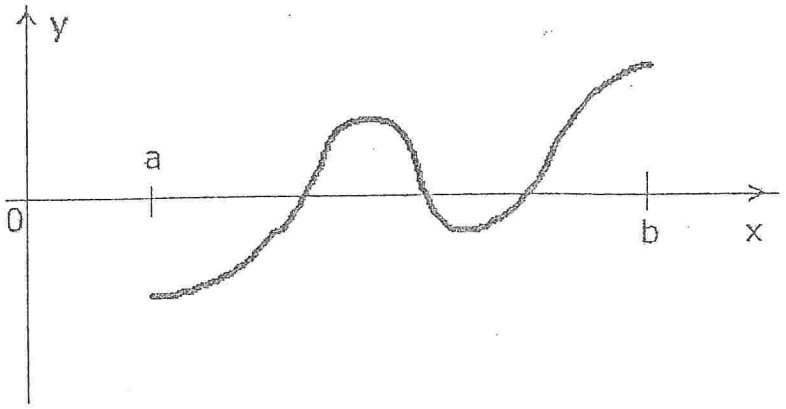

Figura 2.5: Um intervalo contendo raízes de $f(x) = 0$.

Considere então que $f(x)$ é *contínua* em $[a,b]$ e que $f(a)$ e $f(b)$, de acordo com a figura 2.5, apresentam sinais diferentes entre si. Então, nesse caso, podemos afirmar que a curva $y = f(x)$ cruza o eixo das abscissas *ao menos uma vez* em um ponto ξ no intervalo $[a,b]$. Normalmente, podem existir vários desses pontos; porém, se $f(x)$ é *monotônica*, então existe apenas um ponto ξ para o qual $f(\xi) = 0$.

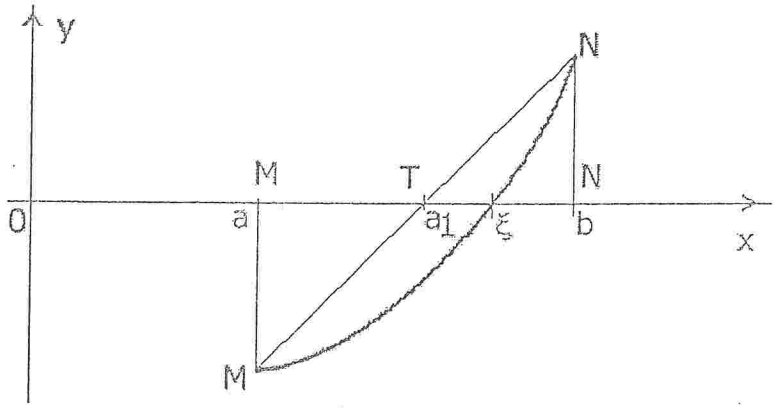

Figura 2.6: Derivação geométrica do método da posição falsa, por triângulos semelhantes.

Considere agora a figura 2.6. Para aproximar o ponto ξ, tome a corda MN e calcule o ponto de interseção T daquela com o eixo das abscissas. Isso pode ser obtido verificando que os triângulos MM_1T e NN_1T são semelhantes; então, podemos escrever

$$\frac{M_1T}{MM_1} = \frac{TN_1}{N_1N}.$$

Observando a figura 2.6, podemos ver que $M_1T = a_1 - a$, $TN_1 = b - a_1$, $MM_1 = -f(a)$ e $N_1N = f(b)$, onde a_1 denota a abscissa do ponto de intersecção T da corda MN com o eixo X. Daí, escrevemos

$$\frac{a_1 - a}{-f(a)} = \frac{b - a_1}{f(b)}$$

e, isolando a_1, temos

$$a_1 = b - f(b)\frac{b-a}{f(b) - f(a)} \tag{2.1}$$

ou, de forma equivalente

$$a_1 = a - f(a)\frac{b-a}{f(b) - f(a)} \tag{2.2}$$

O número a_1 representa o valor aproximado da raiz da equação $f(x) = 0$, situado no intervalo $[a,b]$.

Como, por hipótese, os sinais de $f(a)$ e $f(b)$ são opostos, podemos ter duas situações possíveis:

1. $sign(f(a)) \neq sign(f(a_1))$, ou
2. $sign(f(b)) \neq sign(f(a_1))$.

No primeiro caso, deve-se aplicar a fórmula (2.2) ao intervalo $[a, a_1]$, a fim de se obter a aproximação a_2:

$$a_2 = a - f(a)\frac{a_1 - a}{f(a_1) - f(a)}. \tag{2.3}$$

Se, ao contrário, ocorrer o segundo caso, então obtém-se a_2 através da aplicação da fórmula (2.1) ao intervalo $[a_1, b]$,

$$a_2 = b - f(b)\frac{b - a_1}{f(b) - f(a_1)} \tag{2.4}$$

e assim sucessivamente, sempre verificando o sinal de $f(a_n)$ em relação aos extremos do intervalo em questão. De forma geral, podemos escrever, então:

$$a_{n+1} = a - f(a)\frac{a_n - a}{f(a_n) - f(a)} \tag{2.5}$$

e

$$a_{n+1} = b - f(b)\frac{b - a_n}{f(b) - f(a_n)} \tag{2.6}$$

A figura 2.7 ilustra os casos possíveis com relação à concavidade da curva $y = f(x)$ e os sinais de $f(a)$, $f(b)$ e $f(a_1)$. Se a curva for côncava para cima, deve-se aplicar a equação (2.5) sobre o intervalo $[a, a_1]$ ou a equação (2.6) sobre o intervalo $[a_1, b]$ (figuras 2.7-(a) e 2.7-(d)); se a curva for côncava para baixo, então aplica-se a equação (2.5) sobre o intervalo $[a, a_1]$ ou a equação (2.6) sobre o intervalo $[a_1, b]$ (figuras 2.7-(b) e 2.7-(c)).

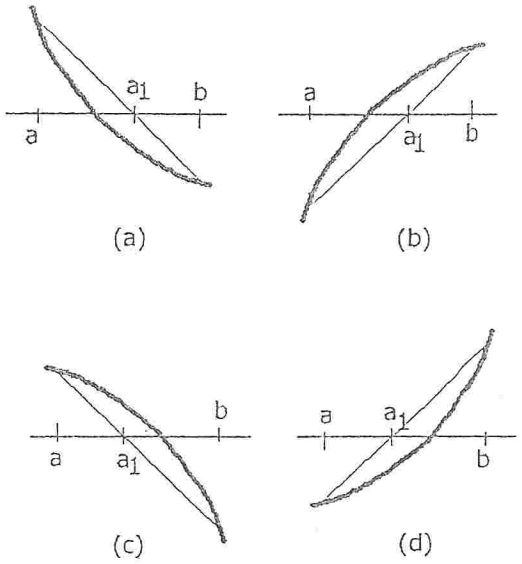

Figura 2.7: Os casos possíveis no método da posição falsa.

Caso não se aplique a equação correta, i.e., desconsiderarmos os sinais de $f(a)$, $f(b)$ e $f(a_n)$, então a_{n+1} pode ficar fora do intervalo $[a, b]$ (ver figura 2.8).

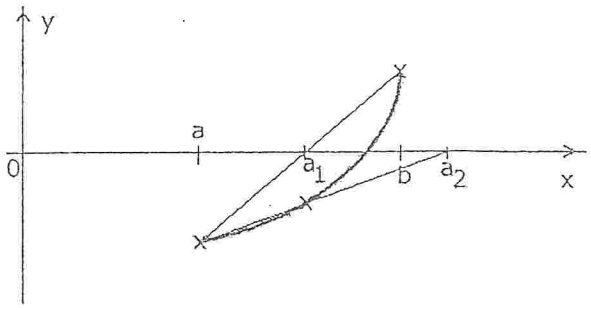

Figura 2.8: O que ocorre quando não se aplica a equação correta no método da posição falsa.

Exemplo 2.2 *Suponha que se deseja obter uma raiz de* $f(x) = x^3 + 3x - 1 = 0$ *no intervalo* $[0, 1]$, *a uma tolerância de* 10^{-3}. *Como* $f(0) = -1$ *e* $f(1) = 3$, *então* $f(x)$ *tem ao menos um zero nesse intervalo. O gráfico da função nesse intervalo (figura 2.9) mostra que ela é côncava para cima e,*

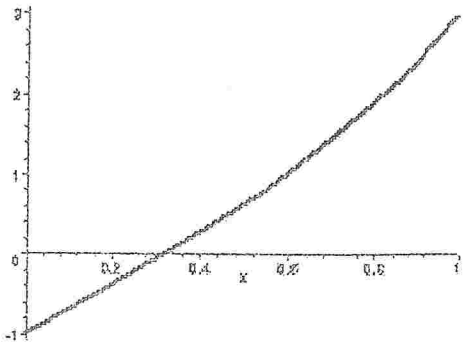

Figura 2.9: Gráfico de $f(x) = x^3 + 3x - 1 = 0$ no intervalo $[0, 1]$.

portanto, devemos usar a equação (2.5). Então, teremos a seguinte seqüência de aproximações:

$$\begin{aligned}
x_1 &= b - f(b)\frac{b-a}{f(b)-f(a)} = 1 - 3\frac{1-0}{3-(-1)} = 0,25 \\
x_2 &= b - f(b)\frac{b-x_1}{f(b)-f(x_1)} = 1 - 3\frac{1-0,25}{3+0,23} = 0,31 \\
x_3 &= 1 - 3\frac{1-0,31}{3+0,040} = 0,319 \\
x_4 &= 1 - 3\frac{1-0,319}{3+0,010} = 0,322 \\
x_5 &= 1 - 3\frac{1-0,322}{3+0,0006} = 0,322
\end{aligned}$$

2.3.1 Melhorando o método da posição falsa

Se analisarmos as equações que governam o método da posição falsa, veremos que elas usam, sempre, um dos extremos – a ou b – do intervalo original.

No entanto, podemos usar as duas últimas aproximações calculadas, pois elas encontram-se mais próximas da raiz; com isso, aumentamos a rapidez com a qual as aproximações convergem para a raiz.

Para tanto, considere a figura 2.10-(a); a fórmula utilizada para se calcular a_{n+1} a partir das duas últimas aproximações é

$$a_{n+1} = a_n - f(a_n)\frac{a_n - a_{n-1}}{f(a_n) - f(a_{n-1})} \tag{2.7}$$

Na figura 2.10-(b), temos um exemplo de uma situação que pode surgir: suponha que a_1 tenha sido calculado através da Equação (2.1) e que a_2 tenha sido calculado através das equações (2.3) ou (2.4). Se, porventura, a_3 é um ponto localizado fora do intervalo original $[a, b]$, então a_3 deve ser substituído por a ou b, antes de se calcular a_4. Particularmente, se $a_3 < a$, então $a_3 \leftarrow a$; se $a_3 > b$, então $a_3 \leftarrow b$.

O método da posição falsa, assim modificado – também conhecido como *método da secante* – apresenta uma taxa de convergência superior ao do método original. Se ξ é a raiz da equação $f(x) = 0$, então

$$|\, a_{n+1} - \xi \,| < C |\, a_n - \xi \,|^t \tag{2.8}$$

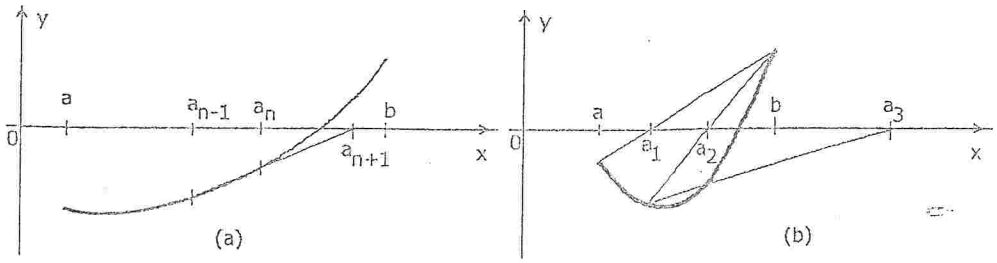

Figura 2.10: Situações possíveis no método modificado da posição falsa.

onde C é uma constante arbitrária, dependente do problema, e t é

$$t = \frac{1+\sqrt{5}}{2} \approx 1,61803398875$$

Exemplo 2.3 *Calcule a raiz da função $x^3 + 3x - 1 = 0$ no intervalo $[0,1]$, a uma tolerância de 10^{-3}.*

Solução: *Calcule $a_1 = 0,2500$ e $a_2 = 0,3100$ com as equações no método da posição falsa original. Após, calcule*

$$\begin{aligned} a_3 &= a_2 - f(a_2)\frac{a_2 - a_1}{f(a_2) - f(a_1)} = \\ &= 0,3100 + 0,0400\frac{0,3100 - 0,2500}{-0,040 + 0,234} = 0,3223 \end{aligned}$$

ou seja, com apenas três aproximações, obtivemos resultado equivalente ao obtido anteriormente.

2.3.2 Análise do erro

Considerando o erro na n-ésima iteração,

$$e_n = x_n - r$$

e assumindo que f'' é contínua e r é uma raiz *simples* de f (i.e., r não é raiz simultânea de f e de f'), substituímos a expressão acima na Equação (2.7):

$$\begin{aligned} e_{n+1} &= x_{n+1} - r = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})} - r = \\ &= \frac{f(x_n)e_{n-1} - f(x_{n-1})e_n}{f(x_n) - f(x_{n-1})} \end{aligned}$$

Fatorando $e_n e_{n-1}$ e multiplicando por $\frac{x_n - x_{n-1}}{x_n - x_{n-1}}$, vem

$$e_{n+1} = \left(\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right)\left(\frac{\frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}}}{x_n - x_{n-1}}\right) e_n e_{n-1}$$

Pelo teorema de Taylor, temos

$$f(x_n) = f(r + e_n) = f(r) + e_n f'(r) + \frac{1}{2}e_n^2 f''(r) + O(e_n^3)$$

e, como $f(r) = 0$,

$$\frac{f(x_n)}{e_n} = f'(r) + \frac{1}{2}e_n f''(r) + O(e_n^2)$$

Escrevendo de forma similar para a $(n-1)$-ésima iteração:

$$\frac{f(x_{n-1})}{e_{n-1}} = f'(r) + \frac{1}{2}e_{n-1}f''(r) + O(e_{n-1}^2)$$

Subtraindo ambas as equações, temos

$$\frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}} = \frac{1}{2}(e_n - e_{n-1})f''(r) + O(e_{n-1}^2)$$

ou, como $x_n - x_{n-1} = e_n - e_{n-1}$ (pela definição de e_n),

$$\frac{\frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}}}{x_n - x_{n-1}} \approx \frac{1}{2}f''(r)$$

Agora, pela definição da derivada em termos do limite, podemos escrever

$$\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \approx \frac{1}{f'(r)}$$

de onde

$$e_{n+1} \approx \frac{1}{2}\frac{f''(r)}{f'(r)}e_n e_{n-1} = Ce_n e_{n-1} \tag{2.9}$$

A Equação (2.9) nos diz que o erro e_{n+1} é proporcional ao produto dos dois erros anteriores; possivelmente, para x_n e x_{n-1} próximos de r, então a taxa de convergência será de ordem quase quadrática.

2.4 Método de Newton-Raphson

O método de Newton é um procedimento genérico que pode ser aplicado em inúmeras situações. Quando temos como problema buscar o zero de uma função real, ele é chamado de método de Newton-Raphson.

Dada então uma aproximação x_0 para a raiz, o método de Newton-Raphson determina uma nova aproximação, x_1, como a intersecção da reta tangente a $f(x_0)$ com o eixo dos x, conforme mostrado na figura 2.11.

Seja então a reta tangente dada por $y = mx + b$, a qual passa pelos pontos $(x_0, f(x_0))$ e $(x_1, 0)$. Assim, para o ponto $(x_1, 0)$, pode-se escrever

$$b = -mx_1 \tag{2.10}$$

e, para o ponto $(x_0, f(x_0))$,

$$\begin{aligned} f(x_0) &= mx_0 - mx_1 \\ x_1 &= x_0 - \frac{f(x_0)}{m} \end{aligned} \tag{2.11}$$

onde m é o coeficiente angular da reta. Esse coeficiente pode ser determinado considerando um outro ponto $(x_0 + h, f(x_0 + h))$, $h \approx 0$, e calculando a reta (secante) que passa pelos pontos $(x_0, f(x_0))$ e $(x_0 + h, f(x_0 + h))$; daí, o coeficiente angular dessa reta é dado por

$$m = \frac{f(x_0 + h) - f(x_0)}{h}. \tag{2.12}$$

Note que, para h suficientemente pequeno, pela definição da derivada de $f(x)$,

$$f'(x) = \lim_{h->0} \frac{f(x + h) - f(x)}{h}$$

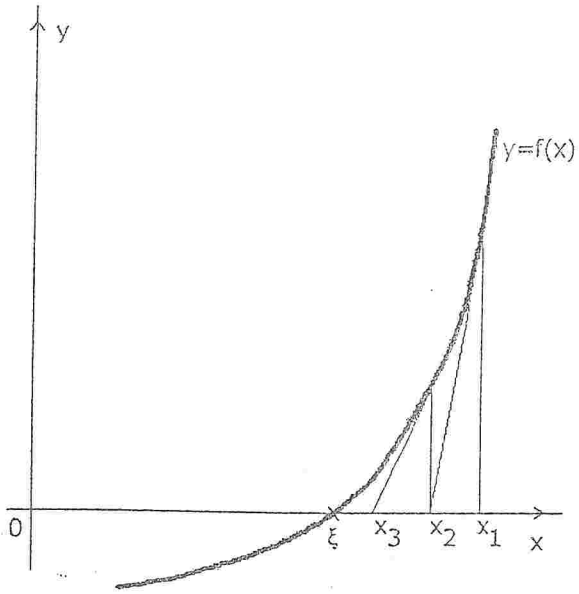

Figura 2.11: O método de Newton-Raphson - interpretação geométrica.

vemos que o coeficiente angular da reta tangente é a própria derivada $f'(x_0)$ (desde que $f'(x)$ exista e seja contínua). Assim, podemos escrever (2.11) como

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{2.13}$$

e, generalizando para uma estimativa x_k,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \ldots \tag{2.14}$$

a qual é a equação governante do método de Newton-Raphson. O método de Newton-Raphson pode ser expresso de forma algorítmica como segue:

Algoritmo 2.4.1 *Newton-Raphson*

proc *newton_raphson*(*input:* x_0, ϵ, δ, $k_{\max}$; *output:* x_{k+1}, k)
  for $k = 0, 1, \ldots, k_{\max}$ *do*
    $x_{k+1} \leftarrow x_k - \frac{f(x_k)}{f'(x_k)}$
    if $|x_{k+1} - x_k| < \delta$ *OR* $|f(x_{k+1})| < \epsilon$ *then*
      break
    endif
  endfor
endproc

Uma outra forma de se obter a equação (2.14) é através de uma expansão de Taylor em torno de uma raiz r. Supondo que x é uma aproximação para r, pelo teorema de Taylor, se f'' existe e é contínua, então

$$\begin{aligned} f(r) &= 0 \\ f(x+h) &= 0 \\ f(x) + hf'(x) + O(h^2) &= 0 \end{aligned}$$

onde $h = r - x$. Se h é pequeno, i.e., x *está próximo de* r, então os termos de ordem igual ou superior a $O(h^2)$ podem ser descartados, e podemos escrever h como

$$h = -\frac{f(x)}{f'(x)} \tag{2.15}$$

Logo, se quisermos corrigir x de forma a aproximá-lo de r, então $x + h = x - \frac{f(x)}{f'(x)}$ é uma aproximação melhor.

O método de Newton-Raphson, no entanto, pode apresentar problemas, dependendo da natureza da função e da estimativa inicial utilizada. Suponha, por exemplo, a figura 2.12. Se x_1 não for tomado suficientemente próximo de r, então a seqüência x_n divergirá.

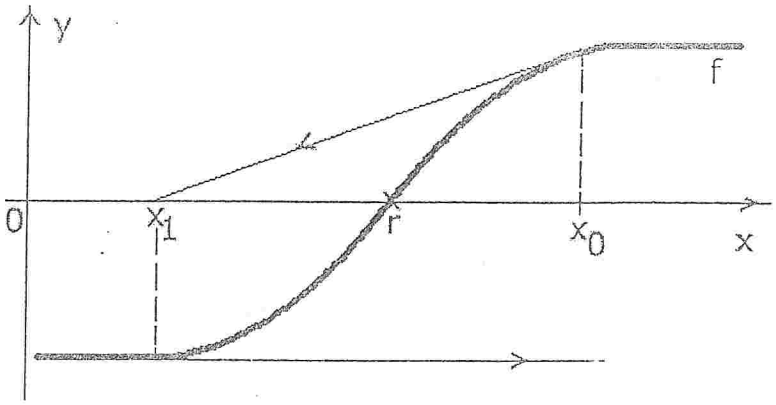

Figura 2.12: Um caso em que a seqüência x_n gerada pelo método de Newton-Raphson diverge.

A aplicação do método de Newton exige um certo cuidado, pois existem algumas situações que podem comprometer o sucesso da sua utilização. Uma situação cujo risco é bastante óbvio é a possibilidade de ocorrer divisão por zero na fórmula iterativa quando $f'(x_i) = 0$. Um bom algoritmo deve checar esta possibilidade, mas é bem possível que quando $f'(x_i)$ está suficientemente próxima de zero, x_i seja uma aproximação aceitável da raiz da equação $f(x) = 0$. Esta situação motiva uma discussão sobre a velocidade de convergência do método de Newton.

Se $\bar{x}$ é uma raiz simples de $f(x) = 0$, o método converge rapidamente e o número de casas decimais exatas praticamente dobra a cada iteração. Por outro lado, se $\bar{x}$ é uma raiz múltipla, o erro em cada aproximação sucessiva é uma fração do erro anterior. Isto é causado pela ordem de aproximação do método, que não é a mesma para os dois casos.

2.4.1 Análise do erro

Considerando o erro na n-ésima iteração,

$$e_n = x_n - r$$

e assumindo que f'' é contínua e r é uma raiz *simples* de f (i.e., r não é raiz simultânea de f e de f'), substituímos a expressão acima na Equação (2.14):

$$\begin{aligned} e_{n+1} &= x_{n+1} - r = x_n - \frac{f(x_n)}{f'(x_n)} - r \\ &= e_n - \frac{f(x_n)}{f'(x_n)} = \frac{e_n f'(x_n) - f(x_n)}{f'(x_n)} \end{aligned}$$

Pelo teorema de Taylor, temos

$$0 = f(r) = f(x_n - e_n) = f(x_n) - e_n f'(x_n) + \frac{1}{2} e_n^2 f''(\xi_n)$$

onde $x_n \leq \xi_n \leq r$. Daí, podemos escrever

$$e_n f'(x_n) - f(x_n) = \frac{1}{2} f''(\xi_n) e_n^2$$

de onde

$$e_{n+1} = \frac{1}{2}\frac{f''(\xi_n)}{f'(\xi_n)} e_n^2 \approx \frac{1}{2}\frac{f''(r)}{f'(r)} e_n^2 = C e_n^2 \tag{2.16}$$

Com base na equação acima, podemos dizer que, se x_n é uma aproximação *suficientemente próxima* de r, então o erro em uma iteração do método de Newton-Raphson decresce de forma proporcional ao quadrado do erro na iteração anterior.

Conforme mencionado acima, o método de Newton pode apresentar problemas, como o caso da divisão por zero. Entretanto, existem outras dificuldades que não são tão facilmente identificáveis. Às vezes, ao invés de as iterações convergirem, elas oscilam para frente e para trás. Isto acontece quando não existem raízes reais (figura 2.13), quando existe simetria em $f(x)$ em torno do ponto $\bar{x}$ (figura 2.14), ou quando a aproximação inicial x_0 está tão longe da raiz correta que alguma outra parte da função acaba interferindo no processo iterativo (figura 2.15).

Um fator importante para a convergência no método de Newton é a escolha do ponto inicial. Dado um intervalo $I = [a, b]$, se $f(a)f''(a) > 0$, $x_0 = a$; se $f(b)f''(b) > 0$, $x_0 = b$. Caso contrário, $x_0 = \frac{a+b}{2}$.

Exemplo 2.4 *Considere o polinômio* $p(x) = x^3 - 5x^2 + 8x - 4$. *Calcule duas raízes utilizando o método de Newton-Raphson.*

Solução: A fórmula iterativa neste caso é

$$x_{i+1} = x_i - \frac{x_i^3 - 5x_i^2 + 8x_i - 4}{3x_i^2 - 10x_i + 8}.$$

Tomando $x_0 = 0$, *obtemos os valores mostrados na tabela 2.1.*

i	x_i
0	0
1	0,5
2	0,8
3	0,95
4	0,995652174
5	0,999962679
6	0,999999998
7	1,0

Tabela 2.1: Valores aproximados da solução pelo método de Newton-Raphson.

Para calcular a raiz seguinte, parte-se de $x_0 = 1,75$, *com o qual obtemos os valores mostrados na tabela 2.2.*

Questão: Por que razão a convergência para a segunda raiz é mais lenta?

O método de Newton pode ser aplicado de maneira um pouco diferente quando a função $f(x)$ é um polinômio, conforme veremos no capítulo 3.

2.5 Derivação numérica

No método de Newton, é necessário utilizar o valor de $f'(x_n)$ a fim de se calcular a nova estimativa x_{n+1}. No entanto, avaliar f' pode ser oneroso (a menos que f seja um polinômio) e, por isso, às vezes recorre-se a uma aproximação numérica de f'.

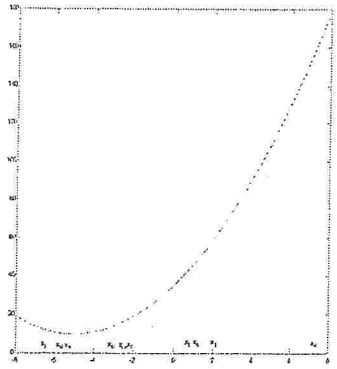

Figura 2.13: Ausência de raízes reais

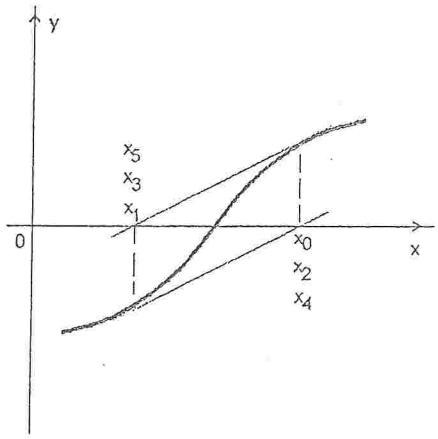

Figura 2.14: Segunda derivada $f''(x^*) = 0$.

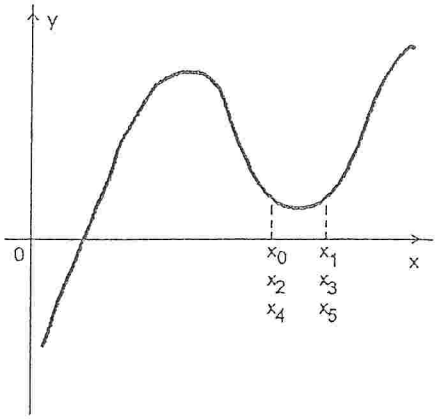

Figura 2.15: Distância inadequada entre x_0 e x^*.

i	x_i
0	1,75
1	1,9
2	1,952941176
3	1,977066274
4	1,988669318
5	1,994367280
6	1,997191745
⋮	
16	1,999999956
17	1,999999956

Tabela 2.2: Valores aproximados da solução pelo método de Newton-Raphson.

Considere a definição da derivada de $f(x)$ em termos do limite,

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{2.17}$$

Se f é *linear* ($f(x) = ax + b$), então a Equação (2.17) é *exata*, i.e., para qualquer $h \neq 0$, ela nos dá o valor correto de $f'(x)$. Se $f(x)$ não for linear, somente em casos muito especiais ela será exata; logo, há um erro envolvido nessa aproximação, o qual pode ser mensurado usando o teorema de Taylor:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(\xi) \tag{2.18}$$

onde $x < \xi < x + h$; f e f' são contínuas em $[x, x+h]$ e f'' é contínua em $(x, x+h)$. Se rearranjarmos os termos da Equação (2.18), obtemos

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(\xi) \tag{2.19}$$

a qual é muito mais útil, pois contém um termo $-\frac{h}{2}f''(\xi)$ – que representa o erro na aproximação da derivada. Dependendo de quanto vale h, o erro tenderá mais ou menos rapidamente para zero.

Exemplo 2.5 *Calcule a derivada de* $\cos x$ *em* $x = \pi/4$, *usando a Equação (2.17), com* $h = 0,01$. *Qual o erro na aproximação?*
Solução: *Em uma HP-48SX, temos*

$$\begin{aligned} f'(x) &\approx \frac{f(x+h) - f(x)}{h} = \frac{0,70000047618 - 0,707106781186}{0,01} \\ &= -0,007106305006 \end{aligned}$$

O erro pode ser estimado como

$$|\frac{h}{2}f''(\xi)| = 0,005|\cos\xi| \leq 0,005$$

Como $\pi/4 < \xi < \pi/4 + h$, *temos* $|\cos\xi| < 0,707106781186$, *de onde obtemos um limitante mais correto para o erro como* $0,0005 \times 0,707106781186 = 0,00353553391$.

Se compararmos com o valor de $f'(\pi/4) = -\text{sen}\frac{\pi}{4} = -0,70710678119$ *em* $x = \pi/4$, *teremos que o erro absoluto é*

$$|-0,70710678119 - (-0,7106305006)| = 0,00352371941$$

o que confirma a nossa estimativa para o erro usando $|\frac{h}{2}f''(\xi)|$.

É óbvio que, para utilizarmos a Equação (2.17), h deve ser pequeno o suficiente. Ora, nessa equação, há a possibilidade de que ocorra perda de dígitos significativos ao calcularmos $f(x+h)-f(x)$, se $f(x+h) \approx f(x)$. Por isso, cuidado deve ser tomado ao se efetuar tais cálculos.

Existem outras aproximações para a derivada de primeira ordem. Suponha, por exemplo, as duas expansões de Taylor, para $f(x+h)$ e $f(x-h)$:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(\xi_1) \tag{2.20}$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\xi_2) \tag{2.21}$$

Subtraindo uma equação da outra, obtemos

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} - \frac{h^2}{12}\left(f'''(\xi_1)+f'''(\xi_2)\right) \tag{2.22}$$

a qual é chamada de aproximação *central* para a derivada de primeira ordem, pois $x-h < x < x+h$. Se assumirmos que $f'''(x)$ existe e é contínua em $[x-h, x+h]$, podemos dizer que $f'''(\xi) = \frac{1}{2}(f'''(\xi_1)+f'''(\xi_2))$, de onde obtemos uma expressão mais simples para o termo envolvendo o erro,

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} - \frac{h^2}{6}f'''(\xi) \tag{2.23}$$

Para se aproximar derivadas de segunda ordem, usualmente se utiliza uma aproximação *central*. Expandindo a série de Taylor nas equações (2.20) e (2.21) por um termo a mais, e somando ambas, obtemos

$$f''(x) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2} - \frac{h^2}{12}f^{(4)}(\xi) \tag{2.24}$$

para $x-h < \xi < x+h$.

2.5.1 O método de Newton-Raphson e as raízes complexas de $f(x)$

O método de Newton-Raphson pode ser utilizado para se extrair as raízes complexas de uma função $f(x)$. Para tanto, basta que se utilize aritmética complexa, tomando cuidado particular com a codificação das funções.

2.6 Exercícios

Exercício 2.1 *Calcule, através do método de Newton-Raphson, todas as raízes da função (inclusive as raízes complexas, se houver)* $f(x) = e^{\operatorname{sen} x} - 2\cos 3x$, *contidas no intervalo* $[-5; 5]$.

Exercício 2.2 *Calcule a raiz positiva de* $f(x) = \frac{(x-1)^2}{x} - 2$ *utilizando os métodos:*

1. *da bissecção*
2. *de Newton-Raphson com aproximação numérica da derivada*
3. *da secante*

Exercício 2.3 *Calcule as intersecções da curva* $y = [(x-1)^2][(x+1)^2]-1/2$ *com o círculo unitário centrado na origem, utilizando o método da secante.*

Exercício 2.4 *Calcule as intersecções das curvas* $f = e^{\frac{1}{(x-1)^3}}$ *e* $g = (x-3)^3+2$, *utilizando o método de Newton-Raphson, com aproximação numérica da derivada.*

Exercício 2.5 *Calcular uma raiz negativa da equação* $f(x) = x^4 - 2x^3 - 6x^2 + 2$ *pelo método da posição falsa.*

Exercício 2.6 *Utilize o método da secante para encontrar o zero positivo de* $f(x) = x - 2\operatorname{sen}(x)$ *com* $x_0 = \pi/2$ *e* $x_1 = 2\pi/3$.

Capítulo 3

Cálculo de Raízes de Polinômios

3.1 Introdução

A determinação de todas, ou de algumas, raízes de um polinômio é um problema importante, o qual tem sido estudado nos últimos quatro séculos. Assim como a fórmula de *Bhaskara* para determinação de raízes de polinômios do segundo grau, existem as fórmulas de *Cardan* e de *Ferrari* para polinômios de terceiro e de quarto grau, respectivamente. Entretanto, foi provado por *Abel*, em 1824, que não existe nenhuma fórmula algébrica finita capaz de calcular as raízes de um polinômio de grau maior ou igual a 5. A partir daí, até hoje, os métodos para o cálculo das n raízes de um polinômio de grau n são voltados aos métodos iterativos, que também podem ser aplicados às equações transcendentais [1].

Os métodos de aproximações sucessivas vistos anteriormente – bissecção, cordas e Newton-Raphson – podem ser utilizados para se determinar uma das raízes de um polinômio; se quisermos todas, então é necessário modificar a função polinomial, através de *deflação*, para os métodos da posição falsa e de Newton-Raphson. Se conhecermos os intervalos em que apenas uma raiz está contida, então podemos usar o método da bissecção para *cada um* dos intervalos.

Além disso, podemos recair no uso de aritmética complexa, pois mesmo um polinômio com coeficientes reais – por exemplo, $z^2 + 1$ – pode ter apenas raízes complexas.

Isso demonstra algumas das dificuldades associadas ao cálculo das raízes de um polinômio. Vejamos então alguns resultados teóricos e métodos específicos para o cálculo de raízes de polinômios.

3.2 Resultados teóricos

Nesta seção, apresentaremos alguns dos teoremas necessários ao entendimento do problema.

Teorema 3.2.1 *Teorema Fundamental da Álgebra: Todo polinômio que não seja uma constante tem ao menos um zero no campo dos números complexos.*

Teorema 3.2.2 *Teorema do Resto: Se um polinômio p, de grau $n \geq 1$, é dividido por um fator $z - c$, então $p(z) = (z - c)q(z) + r$, onde $q(z)$ é o quociente (de grau $n - 1$) e r é um número complexo. Se $z = c$, então $p(c) = r$.*

Teorema 3.2.3 *Teorema dos Fatores: Se um polinômio p, de grau $n \geq 1$, for escrito na forma $p(z) = (z - c)q(z) + r$, e se c for um zero de p, então $r = 0$, de forma que $p(z) = (z - c)q(z)$ e $z - c$ é um fator de p.*

[1] Equações transcendentais são aquelas em que a incógnita aparece submetida à operação não algébrica em pelo menos um termo da função. Ex.: $f(x) = x + cos(x)$, $f(x) = e^{x^2} - sen(x)$

Teorema 3.2.4 *Teorema do Número de Zeros: Um polinômio de grau n tem exatamente n zeros no campo dos números complexos, considerando a multiplicidade de cada zero.*

Teorema 3.2.5 *Teorema do Disco contendo todos os Zeros: Todos os zeros de um polinômio* $p(z) = \sum_{i=0}^{n} a_i z^i$ *encontram-se em um disco fechado cujo centro é a origem do plano complexo e raio*

$$\rho = 1 + \frac{1}{|a_n|} \max_{0 \le k < n} |a_k|.$$

Teorema 3.2.6 *Teorema do Disco contendo todos os Zeros Não-nulos: Se todos os zeros de um polinômio* $s(z) = z^n p(1/z)$ *encontram-se no disco* $\{z : |z| \le \rho\}$, *então todos os zeros não-nulos de p encontram-se fora do disco* $\{z : |z| < \frac{1}{\rho}\}$.

Note que

$$\begin{aligned} s(z) &= z^n \left(a_n \left(\frac{1}{z}\right)^n + a_{n-1} \left(\frac{1}{z}\right)^{n-1} + \ldots + a_0 \right) \\ &= a_n + a_{n-1} z + \ldots + a_0 z^n \end{aligned}$$

O polinômio s tem, também, grau n e os seus coeficientes são os mesmos de p, apenas em ordem reversa. Pode-se verificar que, se $p(z_0) = 0$, então $s(\frac{1}{z_0}) = 0$, para $z_0 \neq 0$.

3.3 Enumeração e localização de raízes de polinômios

Dada uma função $f(x)$, diz-se que $\bar{x}$ é uma raiz ou um zero da equação $f(x) = 0$ se $f(\bar{x}) = 0$. Muitas vezes, não se sabe com certeza quais as raízes de uma determinada função, mas, através de alguns resultados, é possível enumerá-las. Enumerar as raízes de uma função $f(x)$ é dizer quantas raízes ela possui e de que tipo elas são. Se $f(x)$ é um polinômio de grau n, o *teorema fundamental da Álgebra* assegura a existência de n raízes, contando a multiplicidade. Entretanto, responder de que tipo são as raízes (positivas ou negativas, simples ou múltiplas), já não é muito fácil. No caso de funções transcendentais, como não é possível garantir o número de raízes, o problema da sua enumeração acaba por ser mais difícil. Existem algumas regras que permitem enumerar e localizar as raízes de polinômios, conforme mostrado a seguir.

3.3.1 Regra de Descartes

O Teorema 3.3.1 permite obter o número de raízes reais positivas para um polinômio real diferente de zero:

Teorema 3.3.1 *Sejam* $p(z) = a_0 + a_1 z + \ldots + a_n z^n$ *um polinômio real (diferente do polinômio zero), T o número de troca de sinais na seqüência de seus coeficientes* a_k *não nulos, e r o número de suas raízes reais positivas (cada qual contada com a sua respectiva multiplicidade). Então,* $T - r$ *é par e não-negativo.*

Prova Ver [7, pág. 442].

Em outras palavras, a diferença entre o número de trocas de sinal dos coeficientes não-nulos do polinômio e o número de raízes positivas do mesmo é um número par, i.e., $T - r = 2k$ ($k = 0$, $1, \ldots$). Dessa forma, temos $r = T - 2k$, de onde pode-se verificar que o número de raízes reais positivas nunca excede a T.

A mesma regra pode ser aplicada para a enumeração das raízes reais e negativas de $p(z)$, bastando para isso substituir z por $-z$ e, elevando $-z$ às diferentes potências, obter um novo polinômio $q(z) \equiv p(-z)$, cujas raízes positivas são as raízes negativas de $p(z)$.

Exemplo 3.1 *Considere $p(z) = 3z^3 + z^2 - z - 1$. A seqüência de sinais para $p(z)$ é $+ + - -$, de onde $T = 1$; logo, como $r = T - 2k$ e $T - r \geq 0$, pode-se concluir que existe no máximo uma raiz real positiva. Para $p(-z) = -3z^3 + z^2 + z - 1$, temos $T = 2$; de maneira similar, concluímos que o polinômio apresenta ou 2 raízes negativas ou nenhuma.*

Como o Teorema 3.2.1 nos garante que existem 3 raízes para o polinômio em questão, conclui-se que podem existir:

- *Uma raiz real positiva e duas raízes reais negativas;*
- *Uma raiz real positiva, nenhuma raiz real negativa e duas raízes complexas. Lembrando que as raízes complexas de uma equação polinomial com coeficientes reais ocorrem aos pares conjugados, então, como o número máximo de raízes negativas é 2, pode-se concluir que, se não houverem raízes negativas, então necessariamente existe um par de raízes complexas.*

Nesse exemplo, as raízes são: $0,7356705613$ *e* $-0,5345019474 \pm 0,4091564862i$.

Exemplo 3.2 *Considere $p(z) = z^5 - 3z^4 - 2z^3 + z^2 + z + 1$. A seqüência de sinais para $p(z)$ é $+ - - + + +$, de onde $T = 2$; logo, como $r = T - 2k$ e $T - r \geq 0$, pode-se concluir que existe no máximo duas raízes reais positivas. Para $p(-z) = -z^5 - 3z^4 + 2z^3 + z^2 - z + 1$, temos $T = 3$; de maneira similar, concluímos que o polinômio apresenta ou 2 raízes negativas ou nenhuma.*

Como o Teorema 3.2.1 nos garante que existem 5 raízes para o polinômio em questão, conclui-se que podem existir:

- *Duas raízes reais positivas e três raízes reais negativas;*
- *Duas raízes reais positivas, uma raiz real negativa e um par de raízes complexas;*
- *Nenhuma raíz real positiva, três raízes reais negativas e um par de raízes complexas;*
- *Nenhuma raíz real positiva, uma raiz real negativa e dois pares de raízes complexas.*

Nesse exemplo, as raízes são: $3,463105585$; $0,8828320726$; $-0,8675771482$ *e* $-0,2391802550 \pm 0,5666074100i$.

3.3.2 Regra de Du Gua

Seja a equação polinomial $p(z) = a_n z^n + a_1 z^{n-1} + \ldots + a_0 = 0$ de grau n sem raízes nulas. Se, para algum k, $1 \leq k < n$, tem-se $a_k^2 \leq a_{k+1} a_{k-1}$, então $p(z)$ tem raízes complexas.

3.3.3 Regra da lacuna

A regra da lacuna pode ser expressa como segue:

1. Se os coeficientes de $p(z)$ são todos reais e para algum k, $1 \leq k < n$, tem-se $a_k = 0$ e $a_{k-1} a_{k+1} > 0$, então $p(z)$ terá raízes complexas;
2. Se os coeficientes são todos reais e existem dois ou mais coeficientes nulos sucessivos, então $p(z) = 0$ tem raízes complexas.

Exemplo 3.3 *Considere $p(z) = 2z^5 + 3z^4 + z^3 + 2z^2 - 5z + 3$. Para $p(z)$, o número de trocas é $T = 2$, o que implica, pela regra de Descartes, que $p(z)$ tem duas ou zero raízes reais positivas. Para $p(-z)$, o número de trocas é $T = 3$, o que implica que $p(z)$ tem três ou uma raiz real negativa.*

Testando a desigualdade $a_k^2 \leq a_{k+1} a_{k-1}$ para os coeficientes de $p(z) = 0$, tem-se que para $k = 2$, $a_2^2 \leq a_3 a_1$, ou seja, $1 \leq 3 \cdot 2$. Logo, a regra de Du Gua garante a existência de raízes complexas para $p(z)$.

Neste exemplo, a regra da lacuna nada afirma sobre a existência de raízes complexas, pois as condições necessárias não são satisfeitas.

Até aqui, as três regras discutidas não permitem a determinação da localização das raízes. Para estimar o módulo de todas as raízes de um polinômio real $p(z)$, existem as cotas de *Laguerre-Thibault*, de *Fujiwara*, de *Kojima* e de *Cauchy*.

3.3.4 Cota de Laguerre-Thibault

Dado um polinômio $p(z) = 0$, de coeficientes reais, faz-se a divisão de $p(z)$ por $z-1$, $z-2$ e assim sucessivamente, até $z-m$, onde $q(z)$ tenha todos os coeficientes positivos ou nulos, assim como $r > 0$; tal m é chamado de cota superior das raízes reais de $p(z) = 0$. Para determinar a cota inferior, basta fazer o mesmo procedimento para $p(-z)$.

3.3.5 Cota de Fujiwara

Seja $\bar{x}$ uma raiz real ou complexa de $p(z) = a_n z^n + a_{n-1}z^{n-1} + \ldots + a_1 z + a_0 = 0$. Então,

$$|\bar{z}| \leq 2\max\left\{\left|\frac{a_{n-1}}{a_n}\right|, \left|\frac{a_{n-2}}{a_n}\right|^{\frac{1}{2}}, \ldots, \left|\frac{a_1}{a_n}\right|^{\frac{1}{n-1}}, \left|\frac{a_0}{a_n}\right|^{\frac{1}{n}}\right\}.$$

Exemplo 3.4 *Determinar a região do plano onde se encontram as raízes de* $p(z) = z^4 - 14z^2 + 24z - 10$.

A expressão para a cota de Fujiwara fica:

$$\begin{aligned} |\bar{z}| &\leq 2\max\left\{0, 14^{\frac{1}{2}}, 24^{\frac{1}{3}}, 10^{\frac{1}{4}}\right\} \\ &\leq 2 \cdot 3,74 = 7,48 \end{aligned}$$

3.3.6 Cota de Kojima

Dado o polinômio $p(z) = a_n z^n + a_{n-1}z^{n-1} + \ldots + a_1 z + a_0$, toda a raiz $\bar{z}$, real ou complexa, satisfaz

$$|\bar{z}| \leq q_1 + q_2$$

onde q_1 e q_2 são os dois maiores valores de

$$\left\{\left|\frac{a_i}{a_n}\right|^{\frac{1}{i}}\right\}, \quad i = n-1, n-2, \ldots, 0,$$

Exemplo 3.5 *Seja* $p(z) = z^5 + z^4 - 9z^3 - z^2 + 20z - 12$. *Calculando os valores de* $\left\{|a_i/a_0|^{\frac{1}{i}}\right\}$ *para* $i = 4, 3, 2, 1, 0$, *obtém-se o conjunto* $\{1;\ 3;\ 1;\ 2,114742527;\ 1,643751829\}$, *de onde se conclui que* $q_1 = 3$ *e* $q_2 = 2,114742527$. *Logo, toda raiz* $\bar{z}$ *deve satisfazer* $|\bar{z}| < 5,114742527$.

3.3.7 Cota de Cauchy

Toda raiz $\bar{z}$, real ou complexa, de um polinômio real $p(z)$ satisfaz

$$|\bar{z}| < \beta,$$

onde

$$\beta = \lim_{i\to\infty} z_i, \quad \text{com} \quad z_0 = 0,$$

e

$$z_k = \left(\left|\frac{a_{n-1}}{a_n}\right| z_{k-1}^{n-1} + \left|\frac{a_{n-2}}{a_n}\right| z_{k-1}^{n-2} + \ldots + \left|\frac{a_1}{a_n}\right| z_{k-1} + \left|\frac{a_0}{a_n}\right|\right)^{\frac{1}{n}}, \quad k = 0, 1, \ldots$$

Exemplo 3.6 *Estimar a localização das raízes para* $p(z) = z^3 + 2z^2 - 3z - 5$.

Utilizando a cota de Cauchy, tem-se que

$$z_0 = 0, \quad z_{k+1} = \left(2z_i^2 + 3z_i + 5\right)^{\frac{1}{3}},$$

de forma que as iterações resultam

$$\begin{aligned} z_0 &= 0,0 \\ z_1 &= 1,709975 \\ z_2 &= 2,518686 \\ z_3 &= 2,933484 \\ z_4 &= 3,141756 \\ &\vdots \\ z_{14} &= 3,344014 \\ z_{15} &= 3,344095 \end{aligned}$$

Logo, $|\bar{z}| \leq 3,34$.

3.4 Método de Newton-Viéte

O *método de Newton-Viéte* é o método de Newton específico para polinômios, onde o polinômio $p(z)$ e a sua derivada $p'(z)$ são expressos na forma aninhada, ou seja, como

$$p(z) = a_0 + z(a_1 + z(a_2 + \ldots + z(a_n)))\ldots)$$

Nesse caso, a derivada pode ser expressa como

$$p'(z) = a_1 + z(2a_2 + z(3a_3 + \ldots + z(na_n)))\ldots)$$

de onde a fórmula de iteração pode ser escrita como

$$z_{k+1} = z_k - \frac{a_0 + z(a_1 + z(a_2 + \ldots + z(a_n)))\ldots)}{a_1 + z(2a_2 + z(3a_3 + \ldots + z(na_n)))\ldots)}, \quad k = 0, 1, \ldots \tag{3.1}$$

O exemplo a seguir ilustra o funcionamento do método.

Exemplo 3.7 *Encontrar todas as raízes de* $p(z) = z^3 + 2z^2 - 3z - 5$. *Antes de aplicar o algoritmo de Newton-Viéte, é conveniente fazer a enumeração, a localização e a separação das raízes de* $p(z)$.

Enumeração: *1. A regra de Descartes fornece:*

- *para* $p(z) : + + - - \Rightarrow T = 1$;
- *para* $p(-z) : - + + - \Rightarrow T = 2$.

2. A regra da lacuna não pode ser aplicada;

3. A regra de Du Gua nada afirma.

Conclui-se então que $p(z)$ *tem exatamente uma raiz real positiva. As outras duas são ambas reais negativas ou ambas complexas.*

Localização: *A cota de Cauchy é aplicada a partir de*

$$z_0 = 0 \qquad e \qquad z_{k+1} = \left(2z_k^2 + 3x_k + 5\right)^{\frac{1}{3}}, \quad para \quad k = 0, 1, \ldots,$$

o que resulta

$$\begin{aligned} z_1 &= 1,71 \\ z_2 &= 2,52 \\ z_3 &= 2,93 \end{aligned}$$

$$
\begin{aligned}
z_4 &= 3,14 \\
z_5 &= 3,24 \\
z_6 &= 3,29 \\
z_7 &= 3,32 \\
z_8 &= 3,33 \\
z_9 &= 3,34 \\
z_{10} &= 3,34
\end{aligned}
$$

Pode-se dizer que as raízes de $p(z)$ *pertencem à região* $|z| \leq 3,34$.

Separação: *Sabendo que as raízes de* $p(z)$ *estão todas compreendidas na região* $|z| \leq 3,34 < 4$, *constrói-se a seguinte tabela:*

z	-4	-3	-2	-1	0	1	2	3	4
$p(z)$	-25	-5	1	-1	-5	-5	5	31	79

De acordo com a tabela, a raiz real positiva está entre 1 *e* 2; *as raízes reais negativas estão entre* -3 *e* -2 *e entre* -2 *e* -1.

Cálculo das raízes: *As três tabelas a seguir mostram os valores das aproximações das raízes de* $p(z)$. *Na primeira,* $z_0 = -2$, *na segunda,* $z_0 = -1,5$ *e na terceira,* $z_0 = 1,5$. *Logo, as três raízes de* $p(z)$ *são* $\bar{z}_1 = -2,377202854$, $\bar{z}_2 = -1,273890555$ *e* $\bar{z}_3 = 1,651093409$.

i	z_i	$p(z_i)$
0	-2	1
1	-3	-5
2	-2,58333333333	-1,14293981478
3	-2,41242644514	-1,629606481E-01
4	-2,3785447672	-5,97333456E-03
5	-2,37720492776	-9,21691E-06
6	-2,37720285398	-3E-11
7	-2,37720285397	

i	z_i	$p(z_i)$
0	-1,5	0,625
1	-1,22222222222	-1,7146776407E-01
2	-1,27254428341	-4,34794747E-03
3	-1,27388953494	-3,29179E-06
4	-1,27389055496	-2E-11
5	-1,27389055497	

i	z_i	$p(z_i)$
0	1,5	-1,625
1	1,66666666667	1,8518518523E-01
2	1,6512345679	1,66337255E-03
3	1,65109342069	1,3849E-07
4	1,65109340894	4E-11
5	1,65109340894	

O método de Newton-Viéte pode ser utilizado de forma mais eficiente se fizermos uso do método de Horner, o qual será descrito a seguir.

3.5 Método de Horner

O método de Horner, também conhecido como *multiplicação aninhada*, pode ser utilizado não só para se avaliar um polinômio de forma mais eficiente e estável (do ponto de vista numérico), mas também para:

1. Calcular o quociente e o resto de um polinômio dividido por um fator $z - c$;
2. Deflação de um polinômio;
3. Calcular a expansão de Taylor de um polinômio em torno de um ponto.

Vejamos então como proceder a cada um dos cálculos acima.

3.5.1 Cálculo do quociente e do resto

Seja $p(z) = a_0 + a_1 z + \ldots + a_n z^n$, e z_0 um dado número. Então, se escrevermos

$$p(z) = (z - z_0)q(z) + p(z_0) \tag{3.2}$$

temos, pelo Teorema do Resto, que $q(z)$ é um polinômio de grau $n-1$. Esse polinômio pode ser escrito como

$$q(z) = b_0 + b_1 z + \ldots + b_{n-1} z^{n-1}$$

Isolando $q(z)$ na Equação (3.2), e substituindo as expressões para $p(z)$ e $q(z)$, podemos igualar os coeficientes das potências de mesma ordem, de tal forma que obtemos:

$$\begin{aligned} b_{n-1} &= a_n \\ b_{n-2} &= a_{n-1} + z_0 b_{n-1} \\ &\vdots \\ b_0 &= a_1 + z_0 b_1 \\ p(z_0) &= a_0 + z_0 b_0 \end{aligned}$$

ou, em forma compacta, podemos escrever

$$b_{k-1} = a_k + z_0 b_k, \quad k = n-1, n-2, \ldots, 0$$

Um dispositivo que facilita o cálculo dos termos b_k é o seguinte:

	a_n	a_{n-1}	a_{n-2}	$\ldots$	a_0	
z_0		$z_0 b_{n-1}$	$z_0 b_{n-2}$	$\ldots$	$z_0 b_0$	+
	b_{n-1}	b_{n-2}	b_{n-3}	$\ldots$	$p(z_0)$	

Tabela 3.1: *Dispositivo para determinar os coeficientes do polinômio quociente e o resto; os elementos na última linha contém os coeficientes do quociente, bem como o resto, usando como fator* $z - z_0$.

Esse dispositivo pode ser expresso, também, na forma de um algoritmo, chamado de *algoritmo parcial de Horner*, como segue:

Algoritmo 3.5.1 *Horner parcial*

proc horner_parcial(input: n, $[a_0, a_1, \ldots, a_n]$, z_0; *output:* $[b_{-1}, b_0, b_1, \ldots, b_{n-1}]$*)*
  $b_{n-1} \leftarrow a_n$
  for $k = n-1, n-2, \ldots, 0$ *do*
    $b_{k-1} \leftarrow a_k + z_0 b_k$
  endfor
endproc

Note que, ao final do algoritmo, b_{-1} contém o valor de $p(z_0)$. O exemplo a seguir ilustra o uso do procedimento.

Exemplo 3.8 *Se $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$, calcule $p(3)$.*
Solução: *Usando o dispositivo mostrado na Tabela 3.1, temos, para $z_0 = 3$:*

$$\begin{array}{cccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & 12 & 21 \\ \hline & 1 & -1 & 4 & 7 & 19 \end{array} +$$

De onde podemos dizer que $p(3) = 19$ e, ainda, podemos escrever

$$p(z) = (z-3)(z^3 - z^2 + 4z + 7) + 19$$

onde $z^3 - z^2 + 4z + 7 \equiv q(z)$ e $r = 19$.

Exemplo 3.9 *O dispositivo de Horner pode ser usado para se determinar a cota de Laguerre-Thibault (§3.3.4). Considere o polinômio $p(z) = z^3 - 3z^2 - 34z + 120$; nesse caso, ao se dividir $p(z)$ por $z-1$, $z-2$, ..., $z-7$, obtém-se polinômios $q(z)$ com coeficientes negativos. Porém, ao dividí-lo por $z-8$, temos*

$$\begin{array}{cccccc} & 1 & -3 & -34 & 120 \\ 8 & & 8 & 40 & 48 \\ \hline & 1 & 5 & 6 & 168 \end{array} +$$

Pode-se verificar que $q(z)$ tem coeficientes todos positivos e, assim, 8 é uma cota superior para as raízes positivas de $p(z)$. Com efeito, a $p(z)$ tem como raízes -6, 4 e 5.

3.5.2 Deflação de um polinômio

O dispositivo mostrado na subseção anterior pode ser usado para se *remover um fator linear* do polinômio, o que se chama de *deflação.*

Para tanto, basta que z_0 seja tomado como um dos zeros do polinômio; nesse caso, $z - z_0$ é um dos fatores do polinômio (e vice-versa). Os restantes $n-1$ zeros de p são os $n-1$ zeros do polinômio $\frac{p}{z-z_0}$. Vejamos o exemplo a seguir:

Exemplo 3.10 *Se $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$ e 2 é um de seus zeros, deflacione-o adequadamente.*
Solução: *Usando o dispositivo mostrado na Tabela 3.1, temos, para $z_0 = 2$:*

$$\begin{array}{cccccc} & 1 & -4 & 7 & -5 & -2 \\ 2 & & 2 & -4 & 6 & 2 \\ \hline & 1 & -2 & 3 & 1 & 0 \end{array} +$$

De onde podemos dizer que $p(2) = 0$ - conforme esperado - e

$$z^4 - 4z^3 + 7z^2 - 5z - 2 = (z-2)(z^3 - 2z^2 + 3z + 1)$$

e o resto é 0. As três raízes restantes de p devem ser extraídas do polinômio $q(z) = z^3 - 2z^2 + 3z + 1$.

3.5.3 Calcular a expansão de Taylor de um polinômio

Seja

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_2 z^2 + a_1 z + a_0$$

e suponha que se desejam obter os coeficientes c_k na equação

$$\begin{aligned} p(z) &= a_n z^n + a_{n-1} z^{n-1} + \ldots + a_0 \\ &= c_n (z - z_0)^n + c_{n-1} (z - z_0)^{n-1} + \ldots + c_0 \end{aligned}$$

i.e., os coeficientes da expansão de Taylor em torno de z_0. É sabido, obviamente, que esses coeficientes são na forma $c_k = p^{(k)}(z_0)/k!$, mas podemos obtê-los de forma mais eficiente usando o dispositivo de Horner.

Veja que, ao aplicar o dispositivo, obtemos tanto

$$p(z_0) \equiv c_0$$

como

$$q(z) = \frac{p(z) - p(z_0)}{z - z_0} = c_n(z - z_0)^{n-1} + c_{n-1}(z - z_0)^{n-2} + \ldots + c_1$$

o que mostra que c_1 pode ser obtido aplicando o dispositivo de Horner ao polinômio $q(z_0)$, pois $c_1 \equiv q(z_0)$. Pela aplicação sucessiva do dispositivo de Horner aos polinômios quocientes, de graus $n-1, n-2, \ldots, 1$, podemos obter todos os coeficientes da expansão de Taylor, conforme vemos no exemplo abaixo:

Exemplo 3.11 *Se $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$, obtenha a expansão de Taylor em torno de 3.*
Solução: *Usando o dispositivo mostrado na Tabela 3.1, temos, para $z_0 = 3$:*

	1	−4	7	−5	−2	
3		3	−3	12	21	+
	1	−1	4	7	19	
3		3	6	30		+
	1	2	10	37		
3		3	15			+
	1	5	25			
3		3				+
	1	8				

Podemos, então, escrever a expansão de Taylor de $p(z)$ em torno de 3 como

$$p(z) = (z-3)^4 + 8(z-3)^3 + 25(z-3)^2 + 37(z-3) + 19,$$

Esse processo é chamado de *algoritmo completo de Horner*, o qual pode ser expresso da seguinte forma:

Algoritmo 3.5.2 *Horner completo*

```
proc horner_completo(input: n, [a_0, a_1, ..., a_n], z_0; output: [a_0, a_1, ..., a_n])
    for k = 0, 1, ..., n − 1 do
        for j = n − 1, n − 2 ..., 0 do
            a_j ← a_j + z_0 a_{j+1}
        endfor
    endfor
endproc
```

3.5.3.1 O método de Horner e sua relação com a derivada de $p(z)$

Note que, a cada aplicação do algoritmo parcial de Horner, obtém-se um polinômio $q(z)$ e um resto r. Chamemos, agora, de $q_i(z)$ e r_i aos quociente e resto obtidos na i-ésima aplicação do algoritmo.

Se $p(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0$, e z_0 é um número, então após uma aplicação do algoritmo parcial de Horner, temos, conforme já visto,

$$\begin{aligned} q_1(z) &= a_n z^{n-1} + \\ &\quad (a_{n-1} + z_0 a_n) z^{n-2} + \\ &\quad (a_{n-2} + z_0(a_{n-1} + z_0 a_n))\, z^{n-2} + \\ &\quad (a_{n-3} + z_0\,(a_{n-2} + z_0(a_{n-1} + z_0 a_n)))\, z^{n-3} + \ldots \\ r_1 &= a_0 + z_0\,(a_1 + z_0\,(a_2 + z_0\,(a_3 + z_0 a_4))) \end{aligned}$$

onde $q_1(z)$ tem n termos envolvendo a_n e, pela definição do algoritmo, $r_1 \equiv p(z_0)$. Mas quem é $q_1(z)$? O polinômio quociente nada mais é do que o valor da *derivada* de p, avaliada em $z = z_0$. Isto pode ser visto se igualarmos os coeficientes de mesma potência da derivada $p'(z_0)$ e $q_1(z_0)$:

$$\begin{array}{rrcl}
[z_0^{n-1}]: & na_n & = & \sum_{i=1}^{n} a_n = na_n \\
[z_0^{n-2}]: & (n-1)a_{n-1} & = & \sum_{i=1}^{n-1} a_{n-1} = (n-1)a_{n-1} \\
 & & \vdots & \\
[z_0^2]: & 3a_3 & = & \sum_{i=1}^{3} a_3 = 3a_3 \\
[z_0]: & 2a_2 & = & \sum_{i=1}^{2} a_2 = 2a_2 \\
 & a_1 & = & \sum_{i=1}^{1} a_1 = a_1
\end{array}$$

pois existem n termos envolvendo a_n, $n-1$ termos envolvendo a_{n-1}, e assim por diante. Além disso, no i-ésimo termo, existem i produtos envolvendo z_0, o que equivale a z_0^i.

Agora, note que se aplicarmos mais uma vez o algoritmo parcial de Horner, sobre $q_1(z)$, obteremos $q_2(z)$[2] e $r_2 \equiv p'(z_0)$, por analogia. Dessa forma, podemos, aplicando sucessivamente duas vezes o algoritmo parcial de Horner, obter $p(z_0)$ e $p'(z_0)$, de forma bastante econômica e simples. Isso nos leva a obter uma versão modificada do método de Newton-Raphson para o cálculo de uma raiz de um polinômio, conforme descrito a seguir.

3.5.3.2 O método de Newton-Raphson usado em conjunto com o algoritmo parcial de Horner

O método de Newton-Raphson, para se determinar uma raiz de um polinômio, $p(z) = 0$, pode ser expresso por

$$z_{k+1} = z_k - \frac{p(z_k)}{p'(z_k)}, \quad k = 1, 2, \ldots$$

onde z_1 é uma estimativa inicial para a raiz. Ora, como a cada iteração precisamos avaliar p e p' no *mesmo* ponto z_k, é conveniente que combinemos a correção da estimativa z_k com duas aplicações do algoritmo parcial de Horner, usando $z_0 = z_k$.

Os algoritmos a seguir ilustram como combinar de forma efetiva os dois processos:

Algoritmo 3.5.3 *$p(x)$ e $p'(x)$ via Horner parcial*

```
proc horner_parcial_2(input: n, [a_0, a_1, ..., a_n], z_0; output: α, β)
    α ← a_n
    β ← 0
    for k = n − 1, n − 2, ..., 0 do
        β ← α + z_0 β
        α ← a_k + z_0 α
    endfor
endproc
```

[2]Note que $q_2(z)$ será equivalente a $p''(z_0)$ quando avaliarmos $q_2(z_0)$.

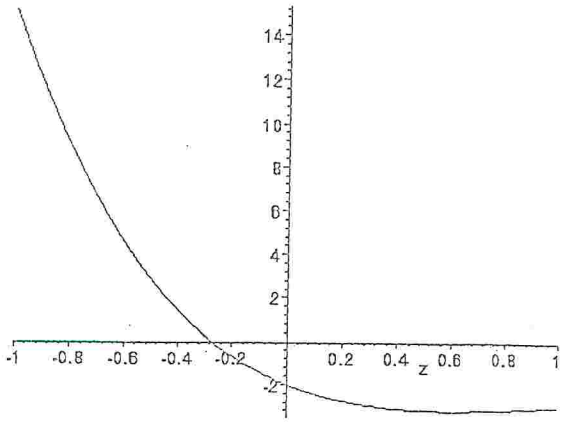

Figura 3.1: Gráfico de $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$ no intervalo $[-1, 1]$.

Algoritmo 3.5.4 *Newton-Raphson para polinômios*

proc newton_raphson_polinômio(input: n, $[a_0, a_1, \ldots, a_n]$, z, ϵ, δ, $k_{\max}$; *output:* z, k)
  $z_0 \leftarrow z$
  call horner_parcial_2(n, $[a_0, a_1, \ldots, a_n]$, z_0; α,β*)*
  for $k = 1, 2, \ldots, k_{\max}$ *do*
    $z \leftarrow z_0 - \frac{\alpha}{\beta}$
    if $|z - z_0| < \delta$ *OR* $|\alpha| < \epsilon$ *then*
      break
    endif
    $z_0 \leftarrow z$
    call horner_parcial_2(n, $[a_0, a_1, \ldots, a_n]$, z_0; α, β*)*
  endfor
endproc

Note que α e β, após a execução do algoritmo *horner_parcial_2*, contém os valores de $p(z_0)$ e $p'(z_0)$. O exemplo abaixo mostra como proceder usando esses dois algoritmos em um problema típico:

Exemplo 3.12 *Para* $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$, *calcule uma raiz de* $p(z) = 0$, *usando como estimativa inicial* $z_1 = 0$.
Solução: *A Figura 3.1 mostra o gráfico do polinômio no intervalo* $[-1, 1]$. *Executando o algoritmo* newton_raphson_polinomio, *obtemos a seguinte seqüência de valores:*

k	$p(z_k)$	$p'(z_k)$	z_k
1	$-2,00000$	$-5,00000$	$-0,40000$
2	$1,40160$	$-12,77600$	$-0,29029$
3	$1,46322$	$-10,17322$	$-0,27591$
4	$0,00226$	$-9,86030$	$-0,27568$
5	$0,00000$	$-9,85537$	$-0,27568$

os quais convergem rapidamente para a raiz $-0,27568$ *naquele intervalo.*

3.6 Raízes complexas de equações polinomiais

A cada par de raízes complexas conjugadas de um polinômio com coeficientes reais $p(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0$ está associado um fator quadrático de $p(z)$ da forma $z^2 - \alpha z - \beta$, onde $\alpha, \beta \in \mathbb{R}$. Se $R = a \pm b\,i$ é uma raiz de $p(z)$, então $\alpha = 2a$ e $\beta = -(a^2 + b^2)$. De maneira geral, $p(z)$ pode ser escrito como

$$p(z) = (z^2 - \alpha z - \beta) q(z) + b_1 (z - \alpha) + b_0, \tag{3.3}$$

onde os termos $b_1(z - \alpha) + b_0$ são o resto da divisão de $p(z)$ por $z^2 - \alpha z - \beta$ e $q(z)$ é um polinômio de grau $n - 2$ que pode ser representado por

$$q(z) = b_n z^{n-2} + b_{n-1} z^{n-3} + \ldots + b_4 z^2 + b_3 z + b_2, \tag{3.4}$$

Desta forma, $p(z)$ fica

$$p(z) = (z^2 - \alpha z - \beta)(b_n z^{n-2} + b_{n-1} z^{n-3} + \ldots + b_3 z + b_2) + b_1 (z - \alpha) + b_0 \tag{3.5}$$

e os termos podem ser expandidos de maneira que

$$\begin{aligned} p(z) &= b_n z^n + (b_{n-1} - \alpha b_n) z^{n-1} + (b_{n-2} - \alpha b_{n-1} - \beta b_n) z^{n-2} + \ldots & (3.6) \\ &+ (b_k - \alpha b_{k+1} - \beta b_{k+2}) z^k + \ldots + (b_1 - \alpha b_2 - \beta b_3) z + b_0 - \alpha b_1 - \beta b_2, & (3.7) \end{aligned}$$

Comparando a equação acima com $p(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0$, chega-se às fórmulas recursivas para o cálculo dos coeficientes b_k de $q(z)$:

$$\begin{aligned} b_n &= a_n \\ b_{n-1} &= a_{n-1} + \alpha b_n \\ b_k &= a_k + \alpha b_{k+1} + \beta b_{k+2} \quad \text{para} \quad k = n-2, n-3, \ldots, 1, 0, \end{aligned} \tag{3.8}$$

O cálculo destes coeficientes também pode ser expresso na forma de uma tabela, semelhante ao visto anteriormente no método de Horner: O exemplo a seguir ilustra o procedimento.

	a_n	a_{n-1}	a_{n-2}	a_{n-3}	$\cdots$	a_k	$\cdots$	a_2	a_1	a_0
β			βb_n	βb_{n-1}	$\cdots$	βb_{k+2}	$\cdots$	βb_4	βb_3	βb_2
α		αb_n	αb_{n-1}	αb_{n-2}	$\cdots$	αb_{k+1}	$\cdots$	αb_3	αb_2	αb_1
	b_n	b_{n-1}	b_{n-2}	b_{n-3}	$\cdots$	b_k	$\cdots$	b_2	b_1	b_0

Exemplo 3.13 *Mostre como dividir* $p(z) = z^5 + 6z^4 - 20z^2 + 22z + 8$ *por* $z^2 + 2z - 3$. *Solução: Neste caso,* $\alpha = -2$ *e* $\beta = 3$. *Montando a tabela, tem-se*

	1	6	0	−20	22	8
3			3	12	−15	6
−2		−2	−8	10	−4	−6
	1	4	−5	2	3	8

Sendo assim, $p(z) = (z^2 + 2z - 3)(z^3 + 4z^2 - 5z + 2) + 3(z + 2) + 8$.

Esta idéia é usada no desenvolvimento do método de Bairstow para o cálculo de coeficientes α e β de tal forma que o fator quadrático $z^2 - \alpha z - \beta$ seja um divisor exato[3] de $p(z)$.

[3] Para que $z^2 - \alpha z - \beta$ seja um divisor exato de $p(z)$, é preciso que $b_1 = b_0 = 0$.

3.6.1 Método de Bairstow

A partir de uma estimativa inicial $z^2 - \alpha_0 z - \beta_0$, $p(z)$ pode ser expresso como

$$p(z) = (z^2 - \alpha_0 z - \beta_0)q(z) + b_1(z - \alpha_0) + b_0, \tag{3.9}$$

Quando b_1 e b_0 são pequenos, $z^2 - \alpha_0 z - \beta_0$ fica próximo de um fator de $p(z)$. Procura-se encontrar então novos valores α_1 e β_1 tal que o fator $z^2 - \alpha_1 z - \beta_1$ fique ainda mais próximo de um fator de $p(z)$. Observa-se que b_0 e b_1 são funções de α e de β, ou seja,

$$b_0 = b_0(\alpha, \beta) \tag{3.10}$$
$$b_1 = b_1(\alpha, \beta). \tag{3.11}$$

Os novos valores α_1 e β_1 satisfazem as relações

$$\alpha_1 = \alpha_0 + \Delta\alpha \tag{3.12}$$
$$\beta_1 = \beta_0 + \Delta\beta \tag{3.13}$$

onde $\Delta\alpha$ e $\Delta\beta$, as correções a serem feitas aos valores de α e de β, são calculadas através da solução do sistema de equações não lineares

$$\begin{cases} b_0(\alpha, \beta) = 0 \\ b_1(\alpha, \beta) = 0 \end{cases} \tag{3.14}$$

Usando o método de Newton para funções de duas variáveis (vide Capítulo 5), tem-se

$$\begin{cases} \Delta\alpha \frac{\partial b_0}{\partial \alpha} + \Delta\beta \frac{\partial b_0}{\partial \beta} = -b_0 \\ \Delta\alpha \frac{\partial b_1}{\partial \alpha} + \Delta\beta \frac{\partial b_1}{\partial \beta} = -b_1 \end{cases} \tag{3.15}$$

onde as derivadas parciais são calculadas em α_0 e em β_0.

Como não é possível expressar b_0 e b_1 explicitamente, como funções de α e de β, as derivadas também não podem ser calculadas explicitamente. Por isto, existem as fórmulas recursivas de Bairstow para calcular numericamente as derivadas parciais.

Para obter $\frac{\partial b_1}{\partial \alpha}$ e $\frac{\partial b_0}{\partial \alpha}$, deriva-se as expressões em (3.8) em relação a α, tendo em mente que os coeficientes a_k são todos constantes e que os b_k são todos funções de α, exceto b_n. Portanto, $\frac{\partial b_n}{\partial \alpha} = 0$ e

$$\begin{aligned}
\frac{\partial b_{n-1}}{\partial \alpha} &= b_n \\
\frac{\partial b_{n-2}}{\partial \alpha} &= b_{n-1} + \alpha \frac{\partial b_{n-1}}{\partial \alpha} \\
\frac{\partial b_{n-3}}{\partial \alpha} &= b_{n-2} + \alpha \frac{\partial b_{n-2}}{\partial \alpha} + \beta \frac{\partial b_{n-1}}{\partial \alpha} \\
&\vdots \\
\frac{\partial b_1}{\partial \alpha} &= b_2 + \alpha \frac{\partial b_2}{\partial \alpha} + \beta \frac{\partial b_3}{\partial \alpha} \\
\frac{\partial b_0}{\partial \alpha} &= b_1 + \alpha \frac{\partial b_1}{\partial \alpha} + \beta \frac{\partial b_2}{\partial \alpha}
\end{aligned}$$

Repetindo o procedimento acima para calcular também as derivadas em relação a β, obtém-se a seguinte relação entre as derivadas parciais:

$$\frac{\partial b_k}{\partial \alpha} = \frac{\partial b_{k-1}}{\partial \beta} \qquad \text{para} \quad k = n, n-1, \ldots, 1, \tag{3.16}$$

Estabelecendo-se que

$$c_{k+1} = \frac{\partial b_k}{\partial \alpha}, \qquad \text{para} \qquad k = 0, 1, \ldots, n-1, \tag{3.17}$$

as equações acima podem ser expressas como

$$c_n = b_n \tag{3.18}$$
$$c_{n-1} = b_{n-1} + \alpha\, c_n \tag{3.19}$$
$$c_k = b_k + \alpha\, c_{k+1} + \beta\, c_{k+2} \qquad \text{para} \qquad k = n-2, n-3, \ldots, 2, 1, \tag{3.20}$$

ou conforme a tabela 3.2.

	a_n	a_{n-1}	a_{n-2}	a_{n-3}	$\cdots$	a_3	a_2	a_1	a_0
β			βb_n	βb_{n-1}	$\cdots$	βb_5	βb_4	βb_3	βb_2
α		αb_n	αb_{n-1}	αb_{n-2}	$\cdots$	αb_4	αb_3	αb_2	αb_1
	b_n	b_{n-1}	b_{n-2}	b_{n-3}	$\cdots$	b_3	b_2	b_1	b_0
β			βc_n	βc_{n-1}	$\cdots$	βc_5	βc_4	βc_3	
α		αc_n	αc_{n-1}	αc_{n-2}	$\cdots$	αc_4	αc_3	αc_2	
	c_n	c_{n-1}	c_{n-2}	c_{n-2}	$\cdots$	c_3	c_2	c_1	

Tabela 3.2: Tabela para cálculo dos coeficientes b_k e c_k.

Com isto, pode-se finalmente formar o sistema

$$\begin{cases} c_1 \Delta\alpha + c_2 \Delta\beta = -b_0 \\ c_2 \Delta\alpha + c_3 \Delta\beta = -b_1 \end{cases} \tag{3.21}$$

o qual deve ser resolvido para determinar os valores α_1 e β_1.

Generalizando, para se calcular um fator $z^2 - \alpha\, z - \beta$ do polinômio $p(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0$ e as raízes correspondentes, executa-se os seguintes passos:

1. Obtém-se uma estimativa $a + bi$ para a raiz (possivelmente através de forma gráfica) e, a partir dessa estimativa, calcula-se $\alpha_0 = 2a$ e $\beta_0 = -(a^2 + b^2)$;
2. Utiliza-se a tabela 3.2 para calcular os coeficientes b_k para $k = 0, 1, \ldots, n$ e c_k para $k = 1, \ldots, n$.
3. Calcula-se, a partir do sistema montado com os coeficientes c_1, c_2 e c_3, as correções $\Delta\alpha$ e $\Delta\beta$ e as novas aproximações α_1 e β_1.
4. Considera-se α_1 como sendo α_0 e β_1 como sendo β_0 e repete-se os passos de 1 a 4 até que ocorra convergência, ou seja, até que $b_0 \approx 0$ e $b_1 \approx 0$.
5. Calcula-se as correspondentes raízes de $p(z)$ a partir da fórmula de Bhaskara.

Esse procedimento pode ser expresso de forma algorítmica conforme expresso nos algoritmos 3.6.1 (o qual calcula os valores de b_0, b_1, c_1, c_2, c_3, conforme a tabela 3.2) e 3.6.2.

Algoritmo 3.6.1 *Algoritmo de Horner quadrático*

proc horner_quadratico(input: a, α, β, n; output: b_0, b_1, c_1, c_2, c_3)
% a contem os coeficientes do polinomio
$b_n \leftarrow a_n$
$b_{n-1} \leftarrow a_{n-1} + \alpha b_n$
for $i \leftarrow n-2, n-3, \ldots, 0$ *do*
$b_i \leftarrow a_i + \alpha b_{i+1} + \beta b_{i+2}$
endfor
$c_n \leftarrow b_n$
$c_{n-1} \leftarrow b_{n-1} + \alpha c_n$
for $i \leftarrow n-2, n-3, \ldots, 1$ *do*
$c_i \leftarrow b_i + \alpha c_{i+1} + \beta c_{i+2}$
endfor
endproc

Algoritmo 3.6.2 *Método de Bairstow*

proc bairstow(input: a, ra, rb, n, $k_{\max}$; output: z_1, z_2)
% a contem os coeficientes do polinomio; ra e rb sao as partes
% real e imaginaria da estimativa da raiz de p(z)
$\alpha \leftarrow 2\,ra$
$\beta \leftarrow -(ra^2 + rb^2)$
$b_0 \leftarrow -1$
$b_1 \leftarrow -1$
for $i \leftarrow 1, 2, \ldots, k_{max}$ *do*
while$(\mid b_0 \mid > 0.0)$ *AND* $(\mid b_1 \mid > 0.0)$ *do*
$(b_0, b_1, c_1, c_2, c_3) \leftarrow$ *horner_quadratico*(a, α, β, n)
Resolve $\begin{bmatrix} c_1 & c_2 \\ c_2 & c_3 \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix} = \begin{bmatrix} -b_0 \\ -b_1 \end{bmatrix}$ *para* $\Delta\alpha$ *e* $\Delta\beta$
$\alpha \leftarrow \alpha + \Delta\alpha$
$\beta \leftarrow \beta + \Delta\beta$
endwhile
endfor
Calcule as raizes z_1 *e* z_2 *da equacao* $z^2 - \alpha z - \beta = 0$
endproc

O método de Bairstow é eficiente, pois ele fornece uma maneira simples de calcular as derivadas parciais requeridas e, além disso, apresenta convergência quadrática. Sua principal deficiência é a dificuldade na escolha dos valores iniciais α_0 e β_0 a fim de garantir convergência.

Exemplo 3.14 *Para* $p(z) = z^4 + z^3 + 3z^2 + 4z + 6$, *considere* $\alpha_0 = -2,1$ *e* $\beta_0 = -1,9$. *Use o método de Bairstow para encontrar* $\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots$, *os fatores quadráticos e as raízes de* $p(z)$.

Solução: A tabela para calcular α_1 *e* β_1 *é*

O sistema linear resultante, que envolve $\Delta\alpha$ *e* $\Delta\beta$ *é*

$$\begin{cases} -12,2740\,\Delta\alpha + 8,2300\,\Delta\beta = -1,7701 \\ 8,2300\,\Delta\alpha - 3,2000\,\Delta\beta = 1,0710 \end{cases} \tag{3.22}$$

A solução deste sistema produz $\Delta\alpha = 0,11069718$ *e* $\Delta\beta = -0,04998819$, *o que implica que os novos valores são*

$$\alpha_1 = -1,98930282 \qquad e \qquad \beta_1 = -1,94998819,$$

	1,0000	1,0000	3,0000	4,0000	6,0000
−1,9000			−1,9000	2,0900	−6,4790
−2,1000		−2,1000	2,3100	−7,1610	2,2491
	1,0000	−1,1000	3,4100	−1,0710	1,7701
−1,9000			−1,9000	6,0800	
−2,1000		−2,1000	6,7200	−17,2830	
	1,0000	−3,2000	8,2300	−12,2740	

A próxima iteração fornece $\alpha_2 = -1,99999277$ *e* $\beta_2 = -2,00015098$, *ou seja, as seqüências estão convergindo para* $\alpha = -2$ *e* $\beta = -2$. *Logo,* $p(z)$ *pode ser fatorado como*

$$p(z) = (x^2 + 2x + 2)(x^2 - x + 3).$$

As quatro raízes complexas, calculadas com a fórmula de Bhaskara, são

$$1 + i, \quad 1 - i, \quad 0,5 + 1,65831239\,i, \quad 0,5 - 1,65831239\,i.$$

3.7 Exercícios

Exercício 3.1 *Aplique a regra de Descartes ao polinômio* $p(z) = 2x^4 - x^3 + 4x^2 - 3x + 7$.

Exercício 3.2 *De que maneira as regras de Du Gua e da lacuna podem ser aplicadas ao polinômio* $p(z) = 2x^4 - x^3 - 3x + 7$?

Exercício 3.3 *Enumerar e localizar as raízes de* $p(z) = 0$, *onde* $p(z) = x^5 + x^4 - 9x^3 - x^2 + 20x - 12$.

Exercício 3.4 *Estimar a localização das raízes para* $p(z) = x^5 + x^4 - 9x^3 - x^2 + 20x - 12$.

Exercício 3.5 *Utilizar o método gráfico e o teorema de Bolzano para estudar as seguintes funções:*

1. $f(x) = x^2 + e^{3x} - 3$
2. $f(x) = e^{-x} - x$
3. $f(x) = \operatorname{sen}(x) - 2e^{-\lambda x}$, *onde* λ *pode variar.*

Exercício 3.6 *Calcular todas as raízes do polinômio* $p(x) = x^3 - x - 1$.

Exercício 3.7 *Calcular todas as raízes de* $p(z) = x^4 - 2x^3 + 4x^2 - 4x + 4$ *iniciando com* $\alpha_0 = 1$ *e* $\beta_0 = -1$.

Exercício 3.8 *Efetue a divisão de* $p(z) = x^5 - 3x^4 + 7x^3 - 10x^2 + 10x - 7$ *por* $x^2 + 2x + 1$.

Capítulo 4

Resolução de Sistemas de Equações Lineares

4.1 Introdução

A resolução de sistemas de equações lineares é um dos problemas numéricos mais comuns em aplicações científicas. Tais sistemas surgem, por exemplo, em conexão com a solução de equações diferenciais parciais, determinação de caminhos ótimos em redes (grafos) e interpolação de pontos, dentre outros.

Consideraremos aqui, inicialmente, a resolução de um sistema de equações lineares de n equações a n variáveis (incógnitas),

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\ddots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

ou, escrito na forma matricial,

$$Ax = b \tag{4.1}$$

onde A é uma matriz quadrada, de ordem n, e x e b são vetores de n elementos,

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

A matriz A pode apresentar, dependendo do problema de onde o sistema foi derivado, uma certa *estrutura* e *esparsidade*. Uma matriz é dita *estruturada* se os seus elementos estão dispostos de uma determinada forma como, por exemplo, ao longo de algumas diagonais e/ou colunas/linhas (figuras 4.1-a) e 4.1-b), como um triângulo (a matriz em 4.1-c é dita *triangular inferior*) ou, ainda, sem estrutura qualquer (4.1-d).

Além disso, as matrizes mostradas na figura 4.1 apresentam alguns elementos nulos. Uma matriz é dita *esparsa* se ela contém, aproximadamente, em torno de 90% de elementos nulos; caso contrário, ela é dita *densa*. Em conseqüência, pode-se dizer que um sistema é *esparso* ou *denso*, dependendo de como é a matriz de coeficientes do sistema.

Uma das principais metas a se atingir, na resolução de um sistema de equações lineares, é obter a sua solução no menor espaço de tempo e, se possível, sem alterar a sua *estrutura* e/ou *esparsidade*. Por isso, existem certos métodos e/ou algoritmos específicos para se resolver alguns sistemas particulares, conforme veremos a seguir.

$$(a)\begin{bmatrix} \times & \times & & \\ \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}, (b)\begin{bmatrix} \times & & & \times \\ & \times & & \times \\ & & \times & \times \\ \times & \times & \times & \times \end{bmatrix},$$

$$(c)\begin{bmatrix} \times & & & \\ \times & \times & & \\ \times & \times & \times & \\ \times & \times & \times & \times \end{bmatrix}, (d)\begin{bmatrix} \times & & \times & \\ \times & \times & & \\ & & \times & \\ \times & & & \times \end{bmatrix}$$

Figura 4.1: *Estruturas típicas de matrizes: (a) tridiagonal, (b) flecha, (c) triangular inferior, (d) não-estruturada.*

4.2 Resolução de Sistemas Triangulares de Equações Lineares

Se o sistema (4.1) apresenta sua matriz de coeficientes A na forma triangular – seja ela *inferior*, como mostrado na figura 4.1-c, ou *superior* – então é possível resolvê-lo de forma imediata, através de *substituição direta*, para matrizes triangulares inferiores, e de *retro-substituição*, para matrizes triangulares superiores.

Suponha então um sistema triangular inferior,

$$Lx = b \tag{4.2}$$

onde

$$L = \begin{bmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix}.$$

Nesse caso, as incógnitas $x_1, x_2, \ldots, x_n$, podem ser facilmente determinadas como

$$\begin{aligned} x_1 &= \frac{b_1}{l_{11}} \\ x_2 &= \frac{b_2 - l_{21}x_1}{l_{22}} \\ x_3 &= \frac{b_3 - l_{31}x_1 - l_{32}x_2}{l_{33}} \\ &\ddots \\ x_n &= \frac{b_n - \sum_{j=1}^{n-1} l_{nj}x_j}{l_{nn}} \end{aligned}$$

O processo acima, denominado de *substituição direta*, pode ser expresso de forma algorítmica como

Algoritmo 4.2.1 *Substituição Direta*

```
proc substituição_direta(input: L, b; output: x)
    for i = 1, 2, ..., n do
        s ← 0
        for j = 1, 2, ..., i − 1 do
            s ← s + l_ij x_j
        endfor
        x_i ← (b_i − s) / l_ii
    endfor
endproc
```

De forma similar, podemos resolver o sistema triangular superior

$$Ux = b \tag{4.3}$$

onde

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ & u_{22} & u_{23} & \dots & u_{2n} \\ & & u_{33} & \dots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{bmatrix}.$$

Nesse caso, as incógnitas x_1, x_2, ..., x_n, podem ser facilmente determinadas como

$$\begin{aligned} x_n &= \frac{b_n}{u_{nn}} \\ x_{n-1} &= \frac{b_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}} \\ x_{n-2} &= \frac{b_{n-2} - u_{n-2,1}x_1 - u_{n-2,2}x_2}{u_{n-2,n-2}} \\ &\ddots \\ x_1 &= \frac{b_1 - \sum_{j=2}^{n} u_{1j}x_j}{u_{11}} \end{aligned}$$

Note que, devido à estrutura de U, as incógnitas são obtidas na ordem contrária àquela com que são obtidas as incógnitas em um sistema triangular inferior. Esse processo é denominado de *retro-substituição* e pode ser expresso de forma algorítmica como

Algoritmo 4.2.2 *Retro-substituição*

```
proc retro_substituição(input: U, b; output: x)
    for i = n, n − 1, ..., 1 do
        s ← 0
        for j = i + 1, i + 2, ..., n do
            s ← s + u_ij x_j
        endfor
        x_i ← (b_i − s) / u_ii
    endfor
endproc
```

4.3 Resolução de Sistemas de Equações Lineares por Eliminação Gaussiana

Se o sistema de equações lineares não apresenta uma forma simples, tal que se possa determinar as incógnitas facilmente, então podemos efetuar modificações no sistema de tal forma que o transformamos em um sistema triangular, preservando a solução do sistema anterior. Uma vez feitas estas modificações, a solução é obtida de forma imediata, conforme visto na seção anterior.

Um processo desse tipo é aquele chamado de *eliminação Gaussiana*, o qual consiste em se aplicar operações elementares – somas e multiplicações – às linhas da matriz de coeficientes e do vetor independente b, de tal forma que a matriz passe a ser triangular superior. Suponha, por exemplo, o sistema

$$\begin{bmatrix} 4 & 2 & 3 \\ -1 & 7 & 3 \\ 4 & 0 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 9 \\ 12 \end{bmatrix}$$

cuja solução é $x_1 = x_2 = x_3 = 1$. Para transformarmos a matriz A em uma matriz triangular superior, devemos eliminar os elementos *abaixo* da diagonal principal de A.

Para tanto, se multiplicamos a primeira linha por $a_{21}/a_{11} = -1/4$ e subtraímo-la da segunda, temos:

$$\begin{bmatrix} 4 & 2 & 3 \\ 0 & 7,5 & 3,75 \\ 4 & 0 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 11,25 \\ 12 \end{bmatrix}$$

Agora, para eliminar o termo a_{31}, multiplicamos a primeira linha por $a_{31}/a_{11} = 1/1$ e subtraímo-la da terceira:

$$\begin{bmatrix} 4 & 2 & 3 \\ 0 & 7,5 & 3,75 \\ 0 & -2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 11,25 \\ 3 \end{bmatrix}$$

Note que os elementos do vetor independente b são modificados também!

A matriz agora é praticamente triangular superior; falta eliminar o termo a_{32}. Para tanto, basta multiplicar a *segunda* linha por $a_{32}/a_{22} = -2/7,5$ e subtraí-la da terceira, de onde

$$\begin{bmatrix} 4 & 2 & 3 \\ 0 & 7,5 & 3,75 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 11,25 \\ 6 \end{bmatrix}$$

Agora, podemos utilizar o algoritmo de retro-substituição para determinar as incógnitas:

$$\begin{aligned} x_3 &= 1 \\ x_2 &= \frac{11,25 - 3,75 \cdot 1}{7,5} = 1 \\ x_1 &= \frac{9 - (2 \cdot 1 + 3 \cdot 1)}{4} = 1 \end{aligned}$$

Podemos sumarizar o processo então da seguinte forma: para se eliminar os elementos abaixo da diagonal na k-ésima coluna (ou seja, os elementos das linhas $k+1$, $k+2$, ..., n na coluna k), usamos o elemento a_{kk} – chamado de *pivô* – para calcularmos um multiplicador $z = \frac{a_{ik}}{a_{kk}}$ para cada i-ésima linha abaixo da linha k. Esse multiplicador será utilizado para multiplicar a k-ésima linha e subtraí-la da linha i (incluindo, aqui, os elementos do termo independente b). Uma vez eliminados todos os elementos abaixo da diagonal principal de A, resta-nos uma matriz triangular superior, e, então, podemos determinar a solução x usando o algoritmo da retro-substituição.

O processo de eliminação Gaussiana pode ser descrito de forma algoritmica como

Algoritmo 4.3.1 *Eliminação Gaussiana*

```
proc eliminação_Gaussiana(input: A, b; output: x)
    for k = 1,2,...,n-1 do
        for i = k+1,k+2,...,n do
            z ← a_ik / a_kk
            a_ik ← 0
            for j = k+1,k+2,...,n do
                a_ij ← a_ij - z a_kj
            endfor
            b_i ← b_i - z b_k
        endfor
    endfor
    call retro_substituição(A,b,x)
endproc
```

4.3.1 Dificuldades

O processo de eliminação Gaussiana, descrito acima, não consegue resolver todo e qualquer sistema. Considere, por exemplo, o sistema

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{4.4}$$

o qual tem como solução $x_1 = x_2 = 1$. No entanto, se formos aplicar eliminação Gaussiana a esse sistema, ele falhará, pois o pivô $a_{11} = 0$. É óbvio, portanto, que os pivôs *não podem ser nulos.*

O sistema (4.4) pode, no entanto, ser modificado, procedendo-se a uma troca de linhas - imediatamente temos um sistema triangular superior. No entanto, é possível que, ao longo do processo de eliminação Gaussiana, surja um zero na diagonal principal e não seja possível, por qualquer troca de linhas, removê-lo. Nesse caso, o sistema não tem solução[1]; o algoritmo para a eliminação Gaussiana deve ser modificado adequadamente para se levar em conta tal possibilidade.

O próximo exemplo mostra uma outra dificuldade associada ao método:

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{4.5}$$

onde $0 < \varepsilon \ll 1$, cuja solução correta é

$$\begin{aligned} x_1 &= \tfrac{1}{1-\varepsilon} \approx 1 \\ x_2 &= \tfrac{1-2\varepsilon}{1-\varepsilon} \approx 1 \end{aligned}$$

No entanto, se aplicarmos eliminação Gaussiana ao sistema (4.5), obteremos

$$\begin{aligned} x_2 &= \tfrac{2-\varepsilon^{-1}}{1-\varepsilon^{-1}} \approx 1 \\ x_1 &= (1-x_2)\varepsilon^{-1} \approx 0 \end{aligned}$$

o qual obviamente aproxima bem x_2, mas o valor de x_1 é completamente errado! Isso acontece porque, se ε é pequeno o suficiente em um determinado computador, tanto $2-\varepsilon^{-1}$ quanto $1-\varepsilon^{-1}$ serão calculados como $-\varepsilon^{-1}$ (devido à perda de dígitos significativos na subtração). Desse exemplo, tiramos uma outra lição: o pivô deve, sempre, ser escolhido como *o maior possível, em módulo.*

O processo de escolha de pivôs, chamado de *pivotamento*, implica na troca de linhas da matriz de coeficientes (bem como do termo independente b). Computacionalmente, no entanto, não é

[1] Esta é, inclusive, uma maneira de se determinar se o sistema é *singular*, isto é, a matriz de coeficientes não tem inversa.

aconselhável se movimentar dados na memória de forma excessiva, pois o tempo de execução do algoritmo passa a ser proibitivo. Podemos, no entanto, modificar o algoritmo de eliminação Gaussiana utilizando um vetor auxiliar de *índices* – chamado de p – o qual implicitamente diz quais linhas foram trocadas; os elementos desse vetor são utilizados para se acessar convenientemente os elementos da matriz e do termo independente. Note, ainda, que o algoritmo deve ser capaz de tratar o caso no qual *não* é necessário se efetuar qualquer troca de linhas.

No algoritmo a seguir, é feito também um *escalonamento* das linhas, isto é, um fator

$$s_i = \max_{1 \le j \le n} |\, a_{ij} \,|, \quad i = 1, 2, \ldots, n$$

é calculado para cada linha. Esse fator é utilizado para se escolher um pivô que seja o maior *relativo* aos elementos de uma coluna; em outras palavras, na k-ésima coluna, iremos selecionar o maior valor $|\, a_{p_i k} \,|/s_{p_i}$ nas linhas $k \le i \le n$.

O algoritmo para a eliminação Gaussiana com pivotamento e escalonamento pode ser expresso como segue:

Algoritmo 4.3.2 *Eliminação Gaussiana com pivotamento e escalonamento*

proc eliminação_Gaussiana_pivotamento_e_escalonamento(*input:* A, b; *output:* x)
  for $i = 1, 2, \ldots, n$ *do*
    $p_i \leftarrow i$
    $s_i \leftarrow \max_{1 \le j \le n} |\, a_{ij} \,|$
  endfor
  for $k = 1, 2, \ldots, n-1$ *do*
    $j \leftarrow k$
    for $i = k+1, k+2, \ldots, n$ *do*
      if $(|\, a_{p_i k} \,|/s_{p_i} \ge |\, a_{p_j k} \,|/s_{p_j})$ *then*
        $j \leftarrow i$
        break
      endif
    endfor
    $q \leftarrow p_k$
    $p_k \leftarrow p_j$
    $p_j \leftarrow q$
    if $(a_{p_k k} = 0)$ *then*
      break
    endif
    for $i = k+1, k+2, \ldots, n$ *do*
      $z \leftarrow \frac{a_{p_i k}}{a_{p_k k}}$
      $a_{p_i k} \leftarrow z$
      for $j = k+1, k+2, \ldots, n$ *do*
        $a_{p_i j} \leftarrow a_{p_i j} - z a_{p_k j}$
      endfor
      $b_{p_i} \leftarrow b_{p_i} - z b_{p_k}$
    endfor
  endfor
  for $i = n, n-1, \ldots, 1$ *do*
    $x_i \leftarrow \left(b_{p_i} - \sum_{j=i+1}^{n} a_{p_i j} x_j \right) / a_{p_i j}$
  endfor
endproc

Note que os fatores $\frac{a_{p_i k}}{a_{p_k k}}$ utilizados para se eliminar os elementos abaixo da diagonal são armazenados na matriz A, onde se colocariam zeros (conforme utilizado no algoritmo da eliminação Gaussiana sem pivotamento). Isso é feito de forma a se poder obter, a partir do algoritmo acima, a fatoração LU da matriz A, conforme veremos na seção a seguir.

O exemplo abaixo mostra o funcionamento do algoritmo descrito acima:

Exemplo 4.1 *Calcule a solução do sistema*

$$\begin{bmatrix} 2 & 3 & -6 \\ 1 & -6 & 8 \\ 3 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -10 \\ 13 \\ 2 \end{bmatrix}$$

Solução: *Inicialmente, temos* $p = (1,2,3)$ *(de acordo com o algoritmo) e* $s = (6,8,3)$ *(verifique, por inspeção). A cada passo, temos:*

k	j	p	i	z
1	3	$(3,2,1)$	2	0,3333
			3	0,6667

$$A = \begin{bmatrix} 0,6667 & 4,3333 & -6,6667 \\ 0,3333 & -5,3333 & 7,6667 \\ 3,0000 & -2,0000 & 1,0000 \end{bmatrix} \quad b = \begin{bmatrix} -11,3333 \\ 12,3333 \\ 2,0000 \end{bmatrix}$$

k	j	p	i	z
2	3	$(3,1,2)$	3	$-1,2308$

$$A = \begin{bmatrix} 0,6667 & 4,3333 & -6,6667 \\ 0,3333 & -1,2308 & -0,5385 \\ 3,0000 & -2,0000 & 1,0000 \end{bmatrix} \quad b = \begin{bmatrix} -11,3333 \\ -1,6154 \\ 2,0000 \end{bmatrix}$$

Uma vez efetuada a eliminação, procede-se ao cálculo das incógnitas:

$$\begin{aligned} i = 3 \quad &: \quad p_3 = 2, x_3 = \frac{-1,6154}{-0,5385} = 3 \\ i = 2 \quad &: \quad p_2 = 1, x_2 = \frac{-11,3333 - (-20)}{4,3333} = 2 \\ i = 1 \quad &: \quad p_1 = 3, x_1 = \frac{2 - (-1)}{3} = 1 \end{aligned}$$

4.3.2 Eliminação Gaussiana e a Fatoração LU

Conforme visto na seção anterior, o algoritmo de eliminação Gaussiana com pivotamento e escalonamento produz, de forma implícita, uma matriz triangular inferior, uma matriz triangular superior e um vetor de permutação. Como essas matrizes foram obtidas por transformações sobre a matriz A original, podemos de alguma forma relacioná-las entre si.

Primeiramente, analisemos o vetor de permutação; seus elementos indicam qual linha foi trocada com outra, i.e., se $p_j = k$, isso significa que a linha j foi trocada com a linha k. Essa permutação pode ser expressa, também, através de uma matriz de permutação, P, a qual tem como elementos apenas o 0 e o 1. No exemplo mostrado na seção anterior, obtemos $p = (3,1,2)$ ao fim; ou seja, a linha 3 está no lugar da linha 1; a linha 1 está no lugar da linha 2 e, por fim, a linha 2 está na linha 3. A matriz de permutação correspondente é

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

A relação existente entre A e as matrizes triangular inferior, L; triangular superior, U; e a matriz de permutação P é a seguinte:

$$PA = LU$$

onde L é triangular inferior com diagonal unitária e os seus elementos abaixo da diagonal principal encontram-se armazenados na matriz A, ao final do algoritmo de eliminação Gaussiana com pivotamento e escalonamento, porém possivelmente permutados.

Usando mais uma vez o exemplo anterior, temos:

$$\begin{array}{rcl} PA & = & LU \\ \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 & -6 \\ 1 & -6 & 8 \\ 3 & -2 & 1 \end{bmatrix} & = & \begin{bmatrix} 1 & 0 & 0 \\ 0,6667 & 1 & 0 \\ 0,3333 & -1,2308 & 1 \end{bmatrix} \begin{bmatrix} 3 & -2 & 1 \\ 0 & 4,3333 & -6,6667 \\ 0 & 0 & -0,5385 \end{bmatrix} \\ & = & \begin{bmatrix} 3 & -2 & 1 \\ 2 & 3 & -6 \\ 1 & -6 & 8 \end{bmatrix} \end{array}$$

onde as matrizes L e U foram permutadas adequadamente, usando a matriz P. Pode-se verificar, por inspeção, que o lado direito da igualdade é a matriz A com as suas linhas trocadas conforme expresso por P.

A fatoração LU é útil quando, para uma mesma matriz de coeficientes A, temos de resolver m sistemas de equações lineares $Ax^{(j)} = b^{(j)}$, com termos independentes $b^{(1)}$, $b^{(2)}$, ..., $b^{(m)}$. Basta, então, obter a fatoração com o algoritmo de eliminação Gaussiana com pivotamento e escalonamento (sem calcular x_i – as últimas três linhas do algoritmo), obtendo L, U e P. Valendo-se da igualdade $PAx = Pb$ e, como $PA = LU$, podemos escrever $L(Ux) = b$, de onde a solução de um sistema $Ax = b$ é obtida resolvendo-se *dois* sistemas triangulares:

$$\begin{array}{rcl} Ly & = & Pb \\ Ux & = & y \end{array}$$

A fatoração LU, bem como a solução dos sistemas triangulares acima, são expressas pelos algoritmos 4.3.3 e 4.3.4.

Algoritmo 4.3.3 *Fatoração LU*

proc fatoração_LU(input: A, b; output: A, b, p)
  for $i = 1, 2, \ldots, n$ *do*
    $p_i \leftarrow i$
    $s_i \leftarrow \max_{1 \leq j \leq n} |\, a_{ij} \,|$
  endfor
  for $k = 1, 2, \ldots, n-1$ *do*
    $j \leftarrow k$
    for $i = k+1, k+2, \ldots, n$ *do*
      if $(|\, a_{p_i k} \,|/s_{p_i} \geq |\, a_{p_j k} \,|/s_{p_j})$ *then*
        $j \leftarrow i$
        break
      endif
    endfor
    $q \leftarrow p_k$
    $p_k \leftarrow p_j$
    $p_j \leftarrow q$
    if $(a_{p_k k} = 0)$ *then*
      break
    endif
    for $i = k+1, k+2, \ldots, n$ *do*
      $z \leftarrow \frac{a_{p_i k}}{a_{p_k k}}$
      $a_{p_i k} \leftarrow z$
      for $j = k+1, k+2, \ldots, n$ *do*
        $a_{p_i j} \leftarrow a_{p_i j} - z a_{p_k j}$
      endfor
    endfor
  endfor
endproc

Algoritmo 4.3.4 *Resolve sistema usando LU*

proc resolve_sistema_LU(input: A, b, p; output: x)
  for $i = 1, 2, \ldots, n$ *do*
    $z_i \leftarrow b_{p_i} - \sum_{j=1}^{i-1} a_{p_i j} z_j$
  endfor
  for $i = n, n-1, \ldots, 1$ *do*
    $x_i \leftarrow \left(z_i - \sum_{j=i+1}^{n} a_{p_i j} x_j \right) / a_{p_i i}$
  endfor
endproc

4.3.3 O Custo Computacional da Fatoração *LU*

O custo computacional de um algoritmo numérico é, normalmente, medido em termos do número de multiplicações e/ou divisões, já que adições e subtrações são efetuadas em uma fração do tempo necessário para aquelas outras duas operações aritméticas. Assim, ao nos referirmos a "operações", estaremos nos referindo a multiplicações e/ou divisões.

Para se obter a fatoração *LU* de uma matriz A, vemos que, quando $k = 1$, no algoritmo respectivo, para cada uma das $n-1$ linhas abaixo da linha 1, é calculado um multiplicador e, então, um múltiplo da primeira linha é subtraído daquelas $n-1$ linhas; isso nos dá n operações.

Como $n-1$ linhas são processadas dessa forma, temos um total de $n(n-1) \approx n^2$ operações para a primeira coluna.

Para as demais colunas, note que o mesmo raciocínio acima é válido, mas é como se a matriz diminuísse de uma linha e uma coluna a cada novo valor de k. Assim, para todos os $n-1$ pivôs a serem calculados, teremos:

$$n^2 + (n-1)^2 + \ldots + 3^2 + 2^2 = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n - 1 \approx \frac{1}{3}n^3 + \frac{1}{2}n^2$$

a qual é obtida usando $\sum_{k=1}^{n} k^2 = \frac{1}{6}n(n+1)(2n+1)$.

Para se corrigir o termo independente b, gasta-se $n-1$ operações, depois $n-2$, e assim sucessivamente, de onde

$$(n-1) + (n-2) + \ldots + 1 = \frac{1}{2}n^2 - \frac{1}{2}n.$$

Finalmente, o processo de retro-substituição custa

$$1 + 2 + 3 + \ldots + n = \frac{1}{2}n^2 + \frac{1}{2}n$$

operações.

Combinando todas as expressões, podemos dizer que, para se resolver m sistemas de equações lineares $Ax^{(i)} = b^{(i)}$, usando a fatoração LU, apresenta um custo computacional de aproximadamente

$$\frac{1}{3}n^3 + \left(\frac{1}{2} + m\right)n^2$$

o que mostra que é mais eficiente efetuar a fatoração LU apenas uma vez, e depois resolver os m sistemas lineares, do que se resolvêssemos cada sistema independentemente, pois o custo, nesse caso, seria da ordem de $\frac{1}{3}mn^3$.

4.3.4 Resolução de sistemas com múltiplos termos independentes

Existem situações que requerem a solução de vários sistemas lineares, todos eles com a *mesma* matriz de coeficientes, porém com *diferentes* termos independentes. Como visto na seção 4.3.3, é mais vantajoso, nesse caso, realizar-se a fatoração LU de A, apenas uma vez; a solução de todos os sistemas é obtida, simplesmente, calculando-se as soluções dos sistemas triangulares $Ly^{(i)} = Pb^{(i)}$ e $Ux^{(i)} = y^{(i)}$, onde o índice (i) identifica um sistema específico.

4.3.4.1 Cálculo da inversa de uma matriz

Uma dessas situações é o cálculo da inversa de uma matriz. Note que tal cálculo *não* é realizado com o fim de se resolver um sistema de equações (utilizando-se a relação $x = A^{-1}b$; aplicações que envolvam certas decomposições de matrizes exigem que se escreva um vetor v como XDX^{-1}, onde X e D são matrizes.

Seja então a matriz A, cuja inversa A^{-1} é desejada. Como, por definição, o produto entre uma matriz e a sua inversa é a matriz identidade I,

$$AA^{-1} = I \tag{4.6}$$

podemos escrever o problema de determinação da inversa na forma de um sistema de equações lineares com múltiplos termos independentes (e, conseqüentemente, múltiplas soluções) como

$$AX = I \tag{4.7}$$

onde $X \equiv A^{-1}$, i.e., as colunas $(X)_i$ são as colunas de A^{-1}.

Dessa forma, obtendo-se a fatoração LU de A, a primeira coluna de A^{-1} é obtida resolvendo-se os sistemas

$$Ly^{(i)} = P \begin{bmatrix} 1 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \tag{4.8}$$

e

$$U(X)_1 = y^{(1)} \tag{4.9}$$

e a segunda coluna é obtida como

$$Ly^{(2)} = P \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 0 & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \tag{4.10}$$

e

$$U(X)_2 = y^{(2)} \tag{4.11}$$

e as demais colunas são obtidas similarmente. Note que, computacionalmente, basta usar apenas um vetor y, sendo o mesmo reutilizado a cada novo sistema resolvido.

Exemplo 4.2 *Obtenha a inversa da matriz*

$$A = \begin{bmatrix} 10 & -10 & 20 \\ -10 & 10 & -10 \\ 20 & -10 & 10 \end{bmatrix}$$

Solução: *Aplicando-se o algoritmo 4.3.3, obtemos os fatores L, U e P:*

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -0,5 & 1 & 0 \\ 0,5 & -1 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 20 & -10 & 10 \\ 0 & 5 & -5 \\ 0 & 0 & 10 \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Agora, aplica-se o algoritmo 4.3.4 usando-se como termo independente o vetor $(1, 0, \ldots, 0)^T$, i.e., resolve-se

$$Ly^{(1)} = P \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -0,5 & 1 & 0 \\ 0,5 & -1 & 1 \end{bmatrix} y^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$y^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

e

$$\begin{aligned} U(X)_1 &= y^{(1)} \\ \begin{bmatrix} 20 & -10 & 10 \\ 0 & 5 & -5 \\ 0 & 0 & 10 \end{bmatrix} (X)_1 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ (X)_1 &= \begin{bmatrix} 0 \\ 0,1 \\ 0,1 \end{bmatrix} \end{aligned}$$

Para a segunda coluna, temos

$$\begin{aligned} Ly^{(2)} &= P \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ -0,5 & 1 & 0 \\ 0,5 & -1 & 1 \end{bmatrix} y^{(2)} &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ y^{(2)} &= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \end{aligned}$$

e

$$\begin{aligned} U(X)_2 &= y^{(2)} \\ \begin{bmatrix} 20 & -10 & 10 \\ 0 & 5 & -5 \\ 0 & 0 & 10 \end{bmatrix} (X)_2 &= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \\ (X)_2 &= \begin{bmatrix} 0,1 \\ 0,3 \\ 0,1 \end{bmatrix} \end{aligned}$$

Finalmente, para a terceira coluna, temos

$$\begin{aligned} Ly^{(3)} &= P \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ -0,5 & 1 & 0 \\ 0,5 & -1 & 1 \end{bmatrix} y^{(3)} &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ y^{(3)} &= \begin{bmatrix} 1 \\ 0,5 \\ 0 \end{bmatrix} \end{aligned}$$

e

$$\begin{aligned} U(X)_3 &= y^{(3)} \\ \begin{bmatrix} 20 & -10 & 10 \\ 0 & 5 & -5 \\ 0 & 0 & 10 \end{bmatrix} (X)_3 &= \begin{bmatrix} 1 \\ 0,5 \\ 0 \end{bmatrix} \\ (X)_3 &= \begin{bmatrix} 0,1 \\ 0,1 \\ 0 \end{bmatrix} \end{aligned}$$

Assim, A^{-1} *é dada por*

$$A^{-1} = X = \left[\begin{bmatrix} 0 \\ 0,1 \\ 0,1 \end{bmatrix} \begin{bmatrix} 0,1 \\ 0,3 \\ 0,1 \end{bmatrix} \begin{bmatrix} 0,1 \\ 0,1 \\ 0 \end{bmatrix}\right]$$

e pode-se verificar que

$$\begin{bmatrix} 10 & -10 & 20 \\ -10 & 10 & -10 \\ 20 & -10 & 10 \end{bmatrix} \begin{bmatrix} 0 & 0,1 & 0,1 \\ 0,1 & 0,3 & 0,1 \\ 0,1 & 0,1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$$AA^{-1} = I$$

4.4 Resolução Iterativa de Sistemas de Equações Lineares

Em certos casos, não é conveniente se resolver o sistema $Ax = b$ através de um método direto como a eliminação Gaussiana. Considere, por exemplo, a matriz A derivada da discretização em diferenças-finitas (com estêncil de 5 pontos) do operador diferencial ∇^2, cuja estrutura é mostrada na figura 4.2; se aplicarmos a fatoração LU sobre A, alguns dos elementos que eram nulos em A passarão a ser diferentes de zero, tanto em L como em U (figura 4.3). Note que A tem 64 elementos não-nulos, ao passo que L e U apresentam um total de 134 elementos não-nulos. A

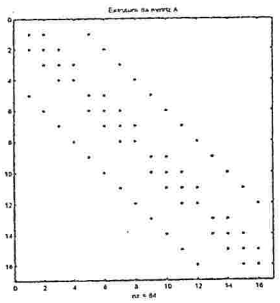

Figura 4.2: *Estrutura da matriz A derivada da discretização em diferenças-finitas do operador diferencial* ∇^2.

eliminação Gaussiana está, nesse caso, *destruindo* a estrutura e/ou a esparsidade da matriz, o que não é aconselhável, principalmente para matrizes grandes ($n > 10000$).

Por outro lado, mesmo quando a matriz é densa – ou seja, a inserção de elementos não-nulos não implicará em aumento considerável do uso da memória – pode não ser aconselhável utilizar um método direto, se a solução desejada necessita apenas um número pequeno de dígitos corretos.

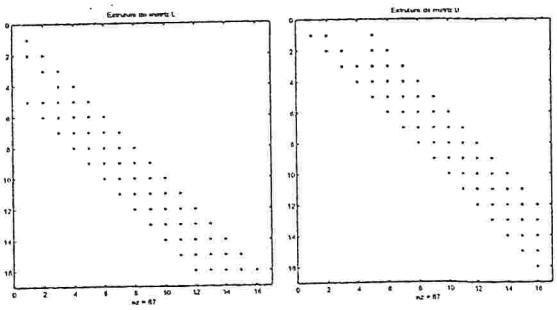

Figura 4.3: *Estruturas da matriz L (à esquerda) e U (à direita), resultantes da fatoração LU de A.*

Uma outra razão, que justifica o uso de métodos *iterativos* (ver [13]), é o fato de seu custo computacional ser proporcional a n^2 (e, às vezes, até mesmo a n), o que os torna bastante competitivos, se comparados a um método direto (cujo custo é proporcional a n^3).

4.4.1 Normas de vetores e de matrizes

Como todo processo iterativo, é necessário saber quando se alcançou a convergência do processo - em nosso caso, obteve-se uma estimativa x_k que aproxima suficientemente $x = A^{-1}b$. Fazendo uma analogia com o método da bissecção (ver seção 2.2), onde se detectava a convergência quando o comprimento do intervalo era menor do que uma tolerância pré-especificada, aqui vamos também calcular um comprimento de um vetor (em $\mathbb{R}^n$).

Para se calcular esse comprimento, utiliza-se uma *norma*. Uma norma de um vetor x pertencente a um espaço vetorial V é uma função $\|x\| : V \longmapsto \mathbb{R}^+$ que obedece aos seguintes postulados:

$$\|x\| > 0, \quad \text{se } x \neq 0, x \in V$$

$$\|\lambda x\| = |\lambda| \|x\|, \quad \text{se } \lambda \in \mathbb{R}, x \in V$$

$$\|x + y\| \leq \|x\| + \|y\| \quad \text{se } x, y \in V (\textit{desigualdade triangular})$$

A norma de um vetor é o seu "comprimento" no espaço vetorial V; é uma generalização da noção de valor absoluto de um número real. Para o espaço vetorial $\mathbb{R}^n$, a norma mais conhecida é a chamada *norma Euclidiana*, definida por

$$\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{\frac{1}{2}} \tag{4.12}$$

onde $x = (x_1, x_2, \ldots, x_n)^T$. Particularmente, em $\mathbb{R}^2$, temos $\|x\|_2 = \sqrt{x_1^2 + x_2^2}$, que é a expressão para a distância de um ponto com coordenadas (x_1, x_2) em relação à origem do sistema de eixos cartesianos.

Existem outras normas que são bastante usadas em cálculos numéricos, como a norma-l_∞

$$\|x\|_\infty = \max_{i=1}^{n} |x_i| \tag{4.13}$$

e a norma-l_1,

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \tag{4.14}$$

as quais são bem mais simples e menos onerosas de se calcular do que a norma Euclidiana.

4.4.2 Normas de matrizes

Uma vez especificada uma norma de um vetor, a *norma matricial subordinada* é definida como

$$\|A\| = \sup \|Au\| : u \in \mathbb{R}^n, \|u\| = 1 \tag{4.15}$$

para uma matriz A $n \times n$. Pode-se verificar que

$$\|Ax\| \leq \|A\| \|x\|, \quad x \in \mathbb{R}^n$$

Por exemplo, a norma matricial subordinada da norma vetorial $\|\cdot\|_\infty$ é dada por

$$\|A\|_\infty = \max_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}| \tag{4.16}$$

4.4.3 Número de condição de uma matriz

Normas de vetores e de matrizes nos permitem avaliar o quão suscetível a erros numéricos será uma computação empregando-se uma dada matriz A. Para tanto, suponha que se deseja resolver o sistema $Ax = b$, onde A é $n \times n$ e A^{-1} existe.

Se A^{-1} tem seus valores perturbados (isto é, ligeiramente modificados), gerando uma nova matriz B, a solução do sistema não é mais $x = A^{-1}b$ mas $\tilde{x} = Bb$. Essa perturbação pode ser medida em termos do comprimento do vetor $x - \tilde{x}$,

$$|| x - \tilde{x} || = || x - Bb || = || x - BAx || = || (I - BA)x || \leq || I - BA || \, || x ||$$

ou

$$\frac{|| x - \tilde{x} ||}{|| x ||} \leq || I - BA ||$$

o que nos dá uma noção do *erro relativo* entre x e $\tilde{x}$.

De forma análoga, suponha que b foi perturbado, gerando um novo vetor $\tilde{b}$. Se x e $\tilde{x}$ são as soluções de $Ax = b$ e $A\tilde{x} = \tilde{b}$, podemos medir o erro absoluto entre x e $\tilde{x}$ escrevendo

$$\begin{aligned} || x - \tilde{x} || &= || A^{-1}b - A^{-1}\tilde{b} || = || A^{-1}(b - \tilde{b}) || \\ &\leq || A^{-1} || \, || b - \tilde{b} || \end{aligned}$$

e o erro relativo como

$$\begin{aligned} || x - \tilde{x} || &\leq || A^{-1} || \, || b - \tilde{b} || = || A^{-1} || \, || Ax || \frac{|| b - \tilde{b} ||}{|| b ||} \\ &\leq || A^{-1} || \, || A || \, || x || \frac{|| b - \tilde{b} ||}{|| b ||} \\ \frac{|| x - \tilde{x} ||}{|| x ||} &\leq || A^{-1} || \, || A || \frac{|| b - \tilde{b} ||}{|| b ||} \end{aligned}$$

o que nos diz que o erro relativo em x é limitado pelo número $|| A^{-1} || \, || A ||$. Essa quantidade é denominada de *número de condição de* A, e é denotada por

$$\kappa(A) = || A^{-1} || \, || A || \tag{4.17}$$

Vejamos um exemplo do uso de $\kappa(A)$.

Exemplo 4.3 *Seja a matriz A e sua inversa A^{-1},*

$$A = \begin{bmatrix} 1 & 1+\varepsilon \\ 1-\varepsilon & 1 \end{bmatrix} \quad A^{-1} = \frac{1}{\varepsilon^2} \begin{bmatrix} 1 & -1-\varepsilon \\ -1+\varepsilon & 1 \end{bmatrix}.$$

Usando a norma-l_∞, então $|| A ||_\infty = 2 + \varepsilon$ e $|| A^{-1} ||_\infty = \frac{1}{\varepsilon^2}(2+\varepsilon)$, de onde

$$\kappa(A) = \left(\frac{2+\varepsilon}{\varepsilon} \right)^2 > \frac{4}{\varepsilon^2}.$$

Se $\varepsilon \leq 0,01$, então $\kappa(A) \geq 40000$. Isso quer dizer que, se b sofrer uma pequena perturbação, a perturbação relativa na solução do sistema $Ax = b$ será 40000 vezes maior!

Uma matriz que tenha um número de condição muito grande é dita *mal-condicionada*, e pequenas variações nos valores de b induzirão um grande erro relativo no vetor solução do sistema.

4.4.4 Erros computacionais e condicionamento

Qualquer solução de um sistema linear deve ser considerada uma solução aproximada, em virtude de erros de arredondamento e outros. O método mais natural para determinação da precisão de uma solução é verificar quão bem esta solução satisfaz o sistema original, calculando o vetor resíduo. Se a solução aproximada $\tilde{x}$ for uma boa aproximação, pode-se esperar que cada componente de $r = b - A\tilde{x}$ seja pequeno, pelo menos em um conceito relativo. Há sistemas de equações, contudo, em que o resto não proporciona uma boa medida da precisão da solução. São sistemas nos quais pequenas alterações nos dados de entrada conduzem a mudanças significativas na solução. Estes são denominados *sistemas instáveis* ou *mal-condicionados.*

Exemplo 4.4 *A solução exata do sistema*

$$\begin{cases} x_1 + x_2 = 2 \\ 1,01x_1 + x_2 = 2,01 \end{cases}$$

é $x_1 = x_2 = 1$. *Supondo que, devido a erros, a solução calculada fosse*

$$\begin{aligned} x_1 &= 0 \\ x_2 &= 2,005 \end{aligned}$$

o vetor resíduo neste caso seria $R^T = [-0,005; 0,005]$. *Entretanto, o erro em cada resposta,* x_1 *e* x_2, *é de aproximadamente uma unidade.*

Por outro lado, os coeficientes também podem conter erros. Supondo que algum tipo de erro tenha mudado as equações acima para

$$\begin{cases} x_1 + x_2 = 2 \\ 1,0001x_1 + x_2 = 2,007 \end{cases}$$

até mesmo uma solução bem diferente da anterior, como $x_1 = 100$ *e* $x_2 = -98$ *produziria um resíduo bem pequeno,* $R^T = [0; -0,003]$.

Erros deste tipo, ao contrário daqueles causados pela acumulação de erros de arredondamento, não podem ser evitados por uma programação cuidadosa. Como, então, determinar quando um problema é mal-condicionado?

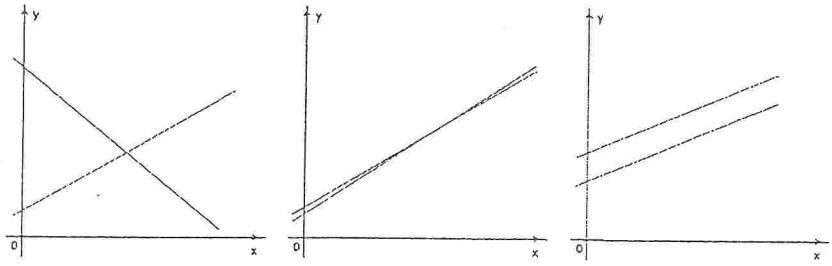

Figura 4.4: *Os sistemas que descrevem a intersecção das retas são, da esquerda para a direita: bem-condicionado, mal-condicionado e singular.*

Em geral, tem-se a situação mostrada na figura 4.4, para o caso de duas retas. Quando o sistema é ordem maior, no entanto, deve-se recorrer a medidas algebricas para se estimar o mal-condicionamento do sistema. Uma dessas medidas é o chamado *determinante normalizado da matriz dos coeficientes.* Para obtê-lo, normaliza-se a matriz de coeficientes A dividindo-se cada

linha de A pela raiz quadrada da soma dos quadrados dos elementos de cada linha,

$$norm|\,A\,| = \begin{vmatrix} \frac{a_{11}}{\alpha_1} & \frac{a_{12}}{\alpha_1} & \cdots & \frac{a_{1n}}{\alpha_1} \\ \frac{a_{21}}{\alpha_2} & \frac{a_{22}}{\alpha_2} & \cdots & \frac{a_{2n}}{\alpha_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n1}}{\alpha_n} & \frac{a_{n2}}{\alpha_n} & \cdots & \frac{a_{nn}}{\alpha_n} \end{vmatrix} = \frac{|\,A\,|}{\alpha_1\alpha_2\cdot\ldots\cdot\alpha_n} \tag{4.18}$$

onde $\alpha_i = \sqrt{a_{i1}^2 + a_{i2}^2 + \ldots + a_{in}^2}$. Diz-se, então, que uma matriz A é mal-condicionada se o número $norm|\,A\,|$ for pequeno, comparado com a unidade.

Exemplo 4.5 *Seja a matriz*

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1{,}01 \end{bmatrix};$$

verifique se ela é mal-condicionada.

Solução: *Calculando-se* $\alpha_1 = \sqrt{1}$ *e* $\alpha_2 = \sqrt{2{,}0201}$, *podemos obter*

$$\text{norm}|\,A\,| = \frac{\begin{vmatrix} 1 & 1 \\ 1 & 1{,}01 \end{vmatrix}}{\alpha_1\alpha_2} \approx 0{,}005$$

ou seja, A é dita ser mal-condicionada.

4.4.5 Métodos iterativos

Dado um sistema não-singular de n equações lineares $Ax = b$, um método iterativo para resolver esse sistema é definido por um conjunto de funções $\Phi_k(x_0, x_1, \ldots, x_k, A, b)$, onde $x_0 = \Phi_0(A, b)$ é uma estimativa inicial para a solução $x = A^{-1}b$ e x_1, x_2, ... são as aproximações sucessivas para a solução,

$$\begin{aligned} x_1 &= \Phi_1(x_0, A, b) \\ x_2 &= \Phi_2(x_0, x_1, A, b) \\ &\vdots \\ x_k &= \Phi_k(x_0, x_1, \ldots, x_k, A, b) \end{aligned}$$

As funções Φ_k nos definem os métodos iterativos. Diz-se que um método é *estacionário* se, para um $m > 0$, Φ_n não depende de n para todo $n \geq m$, ou seja, $\Phi = \Phi_m = \Phi_{m+1} = \ldots$ Nesse caso, x_{n+1} depende de, no máximo, m vetores anteriores, x_n, x_{n-1}, ..., x_{n-m+1}. Por exemplo, para $m = 2$, temos

$$x_0 = \Phi_0(A, b) \tag{4.19}$$

$$x_1 = \Phi_1(x_0, A, b) \tag{4.20}$$

$$x_k = \Phi(x_{k-2}, x_{k-1}, A, b), \quad k = 2, 3, \ldots \tag{4.21}$$

O *grau* de um método estacionário é $\hat{m}$ (para $\hat{m} \leq m$) se, para $n \geq m - 1$, x_{n+1} depende de x_n, x_{n-1}, ..., $x_{n-\hat{m}+1}$ mas *não* para $k < n - \hat{m}+$. O grau de um método iterativo definido pelas equações (4.19)-(4.21) é *2*.

Um método iterativo é dito *linear* se todas as funções Φ_i são funções lineares de x_0, x_1, ..., x_{n-1}. Assim, um *método iterativo estacionário linear de grau 1* pode ser expresso por

$$x_{k+1} = Gx_k + f \tag{4.22}$$

onde G é uma matriz e f um vetor, escolhidos adequadamente.

Para um método como em (4.22), podemos nos referir a um sistema linear relacionado,

$$(I - G)x = f; \tag{4.23}$$

onde I é a matriz identidade de ordem n. Por exemplo, se $G = I - A$, $f \equiv b$, então (4.23) é equivalente a $Ax = b$.

A definição de um método iterativo pode também ser feita a partir de uma *matriz separadora*, Q. Podemos escrever o sistema $Ax = b$ na forma equivalente

$$Qx = (Q - A)x + b \tag{4.24}$$

isto é, $x = \Phi(x, A, b)$, o que nos leva a escrever um processo iterativo, de aproximações sucessivas, como

$$Qx_k = (Q - A)x_{k-1} + b, \quad k = 0, 1, \ldots \tag{4.25}$$

A matriz Q deve ser escolhida de tal forma que se possa calcular rapidamente os x_k e que a seqüência x_0, x_1, ... convirja rapidamente para a solução $x = A^{-1}b$.

A fim de obter uma condição necessária e suficiente para que haja convergência, reescrevemos (4.25) como

$$x_k = (I - Q^{-1}A)x_{k-1} + Q^{-1}b \tag{4.26}$$

A solução x satisfaz a equação

$$x = (I - Q^{-1}A)x_{k-1} + Q^{-1}b \tag{4.27}$$

i.e., x é um *ponto fixo* do mapa

$$x \mapsto (I - Q^{-1}A)x_{k-1} + Q^{-1}b \tag{4.28}$$

Usando as equações (4.26) e (4.27), podemos obter uma expressão para o *erro* $x_k - x$ como

$$x_k - x = (I - Q^{-1}A)(x_{k-1} - x) \tag{4.29}$$

e, aplicando normas, temos

$$\begin{aligned} \| x_k - x \| &\leq \| (I - Q^{-1}A) \| \, \| (x_{k-1} - x) \| \\ &\leq \| (I - Q^{-1}A) \|^2 \, \| (x_{k-2} - x) \| \\ &\leq \vdots \\ &\leq \| (I - Q^{-1}A) \|^k \, \| (x_0 - x) \| \end{aligned} \tag{4.30}$$

de onde

$$\lim_{k \to \infty} \| x_k - x \| = 0, \quad \text{se } \| (I - Q^{-1}A) \| < 1 \tag{4.31}$$

desde que A e Q sejam invertíveis.

4.4.6 Refinamento iterativo

O primeiro método iterativo para a solução de um sistema de equações lineares $Ax = b$ é o chamado *refinamento iterativo*. Para uma estimativa inicial x_0, definimos o vetor *erro* e_0 como

$$e_0 = x - x_0 \tag{4.32}$$

e o vetor *resíduo* r_0 como

$$r_0 = b - Ax_0. \tag{4.33}$$

O vetor erro nos diz o quanto x_0 está distante de x, e o vetor resíduo nos diz o quanto Ax_0 está distante de b.

Multiplicando r_0 por A^{-1}, temos

$$A^{-1}r_0 = A^{-1}b - x_0 = x - x_0 = e_0 \tag{4.34}$$

e, usando essa igualdade, podemos obter uma expressão para x:

$$x = x_0 + A^{-1}r_0 \tag{4.35}$$

Note que a equação (4.35) envolve A^{-1}; mas, obviamente, não podemos utilizá-la, pois se a calculássemos, a solução do sistema seria imediata! Por outro lado, a equação (4.34) nos permite escrever

$$Ae_0 = r_0 \tag{4.36}$$

e, combinando as equações (4.33), (4.36) e (4.35), podemos descrever o *método do refinamento iterativo* como

$$\begin{cases} r_k = b - Ax_k \\ \textit{resolve } Ae_k = r_k \quad , \quad k = 0, 1, \ldots \\ x_{k+1} = x_k + e_k \end{cases} \tag{4.37}$$

O método do refinamento iterativo é utilizado em conjunto com um método direto, como a eliminação Gaussiana. Tendo fatorado A no produto LU e obtido uma solução $\tilde{x}$ para $Ax = b$ (a qual pode não ser muito boa, devido a erros de arredondamento), fazemos $x_0 = \tilde{x}$ e refinamos essa solução, usando (4.37), até que x_k seja suficientemente bom. Note que a fatoração LU pode, agora, ser utilizada para resolver $Ae_k = (LU)e_k = r_k$.

Se consideramos que a solução obtida com a fatoração LU de A não foi exata, então podemos dizer que $U^{-1}L^{-1} = B \approx A^{-1}$. Usando a equação (4.35), escrevemos

$$\begin{aligned} x_{k+1} &= x_k + A^{-1}r_k = x_k + A^{-1}b - A^{-1}Ax_k \therefore B \approx A^{-1} \therefore \\ &= x_k + B(b - Ax_k) \end{aligned} \tag{4.38}$$

De onde podemos mostrar que o método converge para uma solução: subtraindo x de ambos os lados da equação (4.38), temos

$$\begin{aligned} x_{k+1} - x &= x_k - x + B(b - Ax_k) \therefore b = Ax \therefore \\ &= x_k - x + B(Ax - Ax_k) = (I - BA)(x_k - x) \end{aligned}$$

e, tomando normas de ambos os lados da igualdade acima, vem, pela desigualdade triangular:

$$\begin{aligned} ||\, x_{k+1} - x \,|| &\leq ||\, I - BA \,||\, ||x_k - x \,|| \therefore ||\, x_k - x \,|| = ||\, I - BA \,||\, ||\, x_{k-1} - x \,|| \therefore \\ &\leq ||\, I - BA \,||^2 \, ||x_{k-1} - x \,|| \\ &\leq \vdots \\ &\leq ||\, I - BA \,||^k \, ||x_0 - x \,|| \end{aligned}$$

o que nos diz que os erros convergem para 0 se $||\, I - BA \,|| < 1$.

Assim como nos métodos de determinação de raízes de funções, precisamos definir alguns critérios de parada do processo de refinamento. O primeiro desses critérios é a estipulação de um número máximo de iterações ($k_{\max}$); o segundo pode ser baseado na norma do resíduo r_k, devidamente escalonada por $||\, b \,||$ (usando uma norma qualquer, previamente escolhida). Assim, as iterações procederão enquanto

$$||\, r_k \,|| < \varepsilon ||\, b \,|| \tag{4.39}$$

não for satisfeito; ε é um número real, escolhido de acordo com a exatidão requerida.

Um algoritmo que expressa o método do refinamento iterativo pode ser escrito como:

Algoritmo 4.4.1 *Refinamento Iterativo*

```
proc refinamento_iterativo(input: A, L, U, b, x_0, k_max, ε; output: x_k)
    t ← ε||b||
    for k = 0, 1, ..., k_max do
        r_k ← b − Ax_k
        if ||r_k|| < t then
            break
        endif
        resolve (LU)e_k = r_k, obtendo e_k
        x_{k+1} ← x_k + e_k
    endfor
endproc
```

O exemplo abaixo [10] mostra o comportamento desse método.

Exemplo 4.6 *Seja o sistema*

$$\begin{bmatrix} 420 & 210 & 140 & 105 \\ 210 & 140 & 105 & 84 \\ 140 & 105 & 84 & 70 \\ 105 & 84 & 70 & 60 \end{bmatrix} x = \begin{bmatrix} 875 \\ 539 \\ 399 \\ 319 \end{bmatrix}$$

cuja solução é o vetor $x = (1,1,1,1)^T$. *Utilizando um computador com apenas* 6 *casas decimais de precisão, obtemos como solução inicial, através da eliminação Gaussiana, com pivotamento, o vetor*

$$x = (0,999988, 1,000137, 0,999670, 1,000215)^T$$

Agora, dispondo dos fatores triangulares da fatoração LU, podemos utilizar o algoritmo 4.4.1 e obter:

$$x = (0,999994, 1,000069, 0,999831, 1,000110)^T$$

$$x = (0,999996, 1,000046, 0,999891, 1,000070)^T$$

$$x = (0,999993, 1,000080, 0,999812, 1,000121)^T$$

$$x = (1,000000, 1,000006, 0,999984, 1,000011)^T$$

4.4.7 Método iterativo de Jacobi

Suponha o sistema (4.1), com $n = 3$, sem perda de generalidade. Se os elementos da diagonal de A são todos não-nulos, então pode-se isolar cada variável x_1, x_2 e x_3 através de

$$\begin{cases} x_1 = & & c_{12}x_2 + c_{13}x_3 + d_1 \\ x_2 = c_{21}x_1 & & + c_{23}x_3 + d_2 \\ x_3 = c_{31}x_1 + c_{32}x_2 & & + d_3 \end{cases}$$

onde

$$c_{ij} = \begin{cases} -\frac{a_{ij}}{a_{ii}}, & i \neq j \\ 0, & i = j \end{cases}$$

$$d_i = b_i/a_{ii}$$

Com essa transformação, o sistema $Ax = b$ foi transformado em um sistema da forma

$$(I - C)x = d$$

onde $C = D^{-1}(D - A)$, $d = D^{-1}b$ e $D = diag(A)$ (isto é, a matriz formada pelos elementos da diagonal de A). De forma equivalente, podemos escrever

$$x = Cx + d$$

o que sugere uma correção de x por *aproximações sucessivas*,

$$\begin{aligned} x_{k+1} &= Cx_k + d = D^{-1}(D - A)x_k + D^{-1}b = \\ &= (I - D^{-1}A)x_k + D^{-1}b, \quad k = 0, 1, \ldots \end{aligned} \tag{4.40}$$

a qual define o *método iterativo de Jacobi*.

A matriz separadora, aqui, é D^{-1}; o método de Jacobi converge se a matriz A for diagonal dominante, i.e.,

$$|\, a_{ii} \,| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |\, a_{ij} \,| \tag{4.41}$$

e, usando a norma-l_∞,

$$|| \, I - D^{-1}A \, ||_\infty = \max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right|$$

de onde pode-se verificar que a dominância diagonal é condição necessária para a convergência do método.

Para obtermos a solução do sistema $Ax = b$ via o método iterativo de Jacobi, podemos usar a forma equivalente a (4.40),

$$x_{k+1} = x_k - D^{-1}Ax_k + D^{-1}b \tag{4.42}$$

Note que, do ponto de vista de eficiência do processo, deve-se efetuar as divisões de cada linha de A e do elemento respectivo de b pelo elemento na diagonal de A *antes* de se iniciar as iterações. Além disso, se o critério de parada envolve o cálculo do resíduo $r_k = b - Ax_k$, isso exigiria um produto matriz-vetor a mais por iteração, o que pode ser evitado se usarmos como critério de parada $\tilde{r}_{k+1} = D^{-1}r_{k+1}$,

$$|| \, D^{-1}r_{k+1} \, || \leq \varepsilon || \, D^{-1} \, || \, || \, b \, || \tag{4.43}$$

pois

$$\tilde{r}_{k+1} = D^{-1}r_{k+1} = D^{-1}b - D^{-1}Ax_{k+1}$$

e os termos no lado direito da equação já foram calculados, anteriormente, para se obter x_{k+1}. O algoritmo a seguir utiliza essas idéias:

Algoritmo 4.4.2 *Método de Jacobi*

```
proc jacobi(input: A, b, x_0, k_max, ε; output: x_{k+1})
    for i = 1,2,...,n do
        q_i ← a_ii^{-1}
    endfor
    t ← ε||q|| ||b||
    for i = 1,2,...,n do
        for j = 1,2,...,n do
            a_ij ← a_ij * q_i  % sobrescreve A com D^{-1}A
        endfor
        b_i ← b_i * q_i  % sobrescreve b com D^{-1}b
    endfor
    for k = 0,1,...,k_max do
        w ← Ax_k
        x_{k+1} = x_k - w + b
        r̃_{k+1} ← b - w
        if ||r̃_{k+1}|| < t then
            break
        endif
    endfor
endproc
```

O exemplo abaixo ilustra o comportamento típico do método de Jacobi:

Exemplo 4.7 *Resolva o sistema*

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

cuja solução é $x^* = (0,1667, 0,3333, 0,3333, 0,1667)^T$, *usando o método de Jacobi com* $x_0 = (0,0,0,0)^T$ *a uma tolerância* $\varepsilon = 10^{-2}$.

Solução: *Aplicando o método de Jacobi, obtemos*

$$\begin{aligned} x_1 &= (0,0,25, 0,25, 0,25, 0)^T \\ x_2 &= (0,125, 0,25, 0,25, 0,25, 0,125)^T \\ x_3 &= (0,125, 0,3125, 0,3125, 0,3125, 0,125)^T \\ x_4 &= (0,1563, 0,3125, 0,3125, 0,3125, 0,1563)^T \\ x_5 &= (0,1563, 0,3281, 0,3281, 0,3281, 0,1563)^T \\ x_6 &= (0,1641, 0,3281, 0,3281, 0,3281, 0,1641)^T \\ x_7 &= (0,1641, 0,3320, 0,3320, 0,3320, 0,1641)^T \\ x_8 &= (0,1660, 0,3320, 0,3320, 0,3320, 0,1660)^T \end{aligned}$$

ou seja, com oito iterações, obtemos uma aproximação para a solução dentro da tolerância especificada.

4.4.8 Método iterativo de Gauss-Seidel

Analisando o método de Jacobi, vê-se que, a cada iteração, produzem-se todos os elementos do vetor x_{k+1}, usando *apenas* os elementos do vetor x_k. No entanto, nada impede que, à medida

que os elementos de x_{k+1} são produzidos, eles possam ser utilizados para produzir os próximos elementos do *próprio* x_{k+1}. O *método de Gauss-Seidel* faz exatamente isso.

De forma análoga ao método de Jacobi, escrevemos, para $n = 3$,

$$\begin{cases} x_{k+1,1} = & & & u_{12}x_{k,2} + u_{13}x_{k,3} + d_1 \\ x_{k+1,2} = l_{21}x_{k+1,1} & & & + u_{23}x_{k,3} + d_2 \\ x_{k+1,3} = l_{31}x_{k+1,1} + l_{32}x_{k+1,2} & & & + d_3 \end{cases}$$

ou, em forma matricial,

$$x_{k+1} = Lx_{k+1} + Ux_k + d \tag{4.44}$$

onde $L = -D^{-1}A_L$, $U = -D^{-1}A^U$, $D = diag(A)$, $d = D^{-1}b$ e A_L e A_U indicam as porções estritamente *inferior* e *superior* de A (isto é, sem a diagonal).

No caso do método de Gauss-Seidel, podemos escrever a correção para x_{k+1} de forma mais compacta; note que a expressão $Lx_{k+1} + Ux_k$ pode ser calculada como

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} \left(\frac{a_{ij}}{a_{ii}}\right) x_j, \quad i = 1, 2, \ldots, n$$

O critério de parada, no entanto, deve ser calculado usando o resíduo $r_{k+1} = b - Ax_{k+1}$, como mostra o algoritmo 4.4.3.

Algoritmo 4.4.3 *Método de Gauss-Seidel*

```
proc gauss_seidel(input: A, b, x_0, k_max, ε; output: x_{k+1})
    for i = 1,2,...,n do
        q_i ← a_ii^{-1}
    endfor
    t ← ε||q|| ||b||
    for i = 1,2,...,n do
        for j = 1,2,...,n do
            a_ij ← a_ij * q_i  % sobrescreve A com D^{-1}A
        endfor
        b_i ← b_i * q_i  % sobrescreve b com D^{-1}b
    endfor
    for k = 0,1,...,k_max do
        u ← x_k
        for i = 1,2,...,n do
            u_i ← b - Σ_{j=1, j≠i}^{n} a_ij u_j
        endfor
        x_{k+1} ← u
        r_{k+1} ← b - A x_{k+1}
        if ||r̄_{k+1}|| < t then
            break
        endif
    endfor
endproc
```

O exemplo 4.8 ilustra o comportamento típico do método de Gauss-Seidel:

Exemplo 4.8 *Resolva o sistema*

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

cuja solução é $x^* = (0,1667, 0,3333, 0,3333, 0,1667)^T$, *usando o método de Gauss-Seidel com* $x_0 = (0,0,0,0)^T$ *a uma tolerância* $\varepsilon = 10^{-2}$.

Solução: *Aplicando o método de Gauss-Seidel, obtemos*

$$\begin{aligned} x_1 &= (0,0,25, 0,25, 0,125)^T \\ x_2 &= (0,125, 0,3125, 0,3125, 0,1563)^T \\ x_3 &= (0,1563, 0,3281, 0,3281, 0,1641)^T \\ x_4 &= (0,1641, 0,3320, 0,3320, 0,1660)^T \end{aligned}$$

ou seja, com quatro iterações, obtemos uma aproximação para a solução dentro da tolerância especificada.

Da mesma forma que o método de Jacobi, uma condição necessária e suficiente para a convergência do método de Gauss-Seidel é que a matriz A seja diagonal-dominante (ver equação 4.41). Existe um critério – de *Sassenfeld* – que, se atendido, garante a convergência do método. Para se verificar se uma matriz de coeficientes do sistema satisfaz a tal critério, calcula-se os valores S_1, $S_2, \ldots, S_n$, definidos como

$$\begin{aligned} S_1 &= \tfrac{1}{|a_{11}|}(|a_{12}| + |a_{13}| + \ldots + |a_{1n}| \\ S_2 &= \tfrac{1}{|a_{22}|}(|a_{21}|S_1 + |a_{23}| + \ldots + |a_{2n}| \\ &\vdots \\ S_n &= \tfrac{1}{|a_{nn}|}(|a_{n1}|S_1 + |a_{n2}|S_2 + \ldots + |a_{nn-1}|S_{n-1} \end{aligned} \qquad (4.45)$$

e, se

$$S_i < 1, \quad \forall\, 1 \leq i \leq n$$

então o método de Gauss-Seidel irá convergir.

Exemplo 4.9 *Para a matriz do exemplo 4.8, verifique se o critério de Sassenfeld é atendido.*

Solução: *Calculando os valores de* S_i, *temos:*

$$\begin{aligned} S_1 &= \frac{1}{|4|}(|-1| + |-1|) = 0,5 < 1 \\ S_2 &= \frac{1}{|4|}(|-1|0,5 + |-1|) = 0,375 < 1 \\ S_3 &= \frac{1}{|4|}(|-1|0,5 + |-1|) = 0,375 < 1 \\ S_4 &= \frac{1}{|4|}(|-1|0,375 + |-1|0,375) = 0,1875 < 1 \end{aligned}$$

e, como $S_i < 1$, $1 \leq i \leq 4$, *o critério de Sassenfeld é atendido e, por conseguinte, o método de Gauss-Seidel é convergente para um sistema com essa matriz de coeficientes.*

O critério de Sassenfeld é, no entanto, apenas suficiente; uma matriz pode não atendê-lo e, mesmo assim, o método de Gauss-Seidel pode convergir, como mostra o exemplo abaixo.

Exemplo 4.10 *Resolva o sistema*

$$\begin{bmatrix} 1 & 1 \\ 1 & -3 \end{bmatrix} x = \begin{bmatrix} 3 \\ -3 \end{bmatrix}$$

Solução: *O critério de Sassenfeld não é satisfeito pois, calculando os valores de* S_i, *temos*

$$\begin{aligned} S_1 &= \frac{1}{|1|}(|1| = 1 \not< 1 \\ S_2 &= \frac{1}{|-3|}(|1|1) = 0,333\ldots < 1 \end{aligned}$$

No entanto, o método de Gauss-Seidel converge para a solução $x^* = (1,5,1,5)^T$ *em* 11 *iterações, a uma tolerância de* 10^{-4}*:*

$$\begin{aligned}
x_0 &= (0,0000, 0,0000)^T \\
x_1 &= (3,0000, 2,0000)^T \\
x_2 &= (1,0000, 1,3333)^T \\
x_3 &= (1,6667, 1,5556)^T \\
x_4 &= (1,4444, 1,4815)^T \\
x_5 &= (1,5185, 1,5062)^T \\
x_6 &= (1,4938, 1,4979)^T \\
x_7 &= (1,5021, 1,5007)^T \\
x_8 &= (1,4993, 1,4998)^T \\
x_9 &= (1,5002, 1,5001)^T \\
x_{10} &= (1,4999, 1,5000)^T \\
x_{11} &= (1,5000, 1,5000)^T
\end{aligned}$$

Note que a dominância diagonal de uma matriz é relacionada com a *ordem* em que as equações se apresentam. Uma simples troca entre duas linhas pode ser desastrosa, como mostra o exemplo a seguir.

Exemplo 4.11 *Seja o sistema apresentado no exemplo 4.10, com as linhas trocadas entre si, i.e.*

$$\begin{bmatrix} 1 & -3 \\ 1 & 1 \end{bmatrix} x = \begin{bmatrix} -3 \\ 3 \end{bmatrix}$$

Nesse caso, como a matriz do sistema não é diagonal-dominante, o método de Gauss-Seidel diverge, apresentando como primeiras estimativas os vetores

$$\begin{aligned}
x_0 &= (0,0000, 0,0000)^T \\
x_1 &= (-3,0000, 6,0000)^T \\
x_2 &= (15,0000, -12,0000)^T \\
x_3 &= (-39,0000, 42,0000)^T \\
&\vdots \\
x_8 &= (-29523,0000, 29526,0000)^T \\
x_9 &= (88575,000, -88572,0000)^T \\
&\vdots \\
x_{20} &= (5,2302 \times 10^9, -5,2302 \times 10^9)^T
\end{aligned}$$

apesar do sistema ter a mesma solução $x^* = (1,5,1,5)^T$.

4.4.9 Extrapolação de um método iterativo

Uma das formas de garantir e/ou acelerar a convergência de um método iterativo é utilizar uma técnica de *extrapolação*, a qual consiste em se combinar a correção da estimativa x_k – dada pela equação governante do método iterativo – com uma outra correção, semelhante. Em termos das funções Φ, isso pode ser expresso como

$$x_{k+1} = \omega\Phi_k(x_0, x_1, \ldots, x_k, A, b) + (1-\omega)\bar{\Phi}_k(x_0, x_1, \ldots, x_k, A, b), \omega \in \mathbb{R} \qquad (4.46)$$

Note que, se $\omega = 1$, temos o método iterativo original.

Por exemplo, no caso do método de Jacobi, podemos usar $\bar{\Phi}_k(x_0, x_1, \ldots, x_k, A, b) = I$ (ou seja, a matriz identidade). Assim temos o *método de relaxação de Jacobi* (JOR),

$$x_{k+1} = \omega(x_k - D^{-1}Ax_k + D^{-1}b) + (1-\omega)x_k, \quad 0 < \omega \leq 1 \tag{4.47}$$

e, de forma análoga, temos o *método das relaxações sucessivas* (SOR), uma variante do método de Gauss-Seidel,

$$x_{k+1} = \omega(Lx_{k+1} + Ux_k + d) + (1-\omega)x_k, \quad 0 < \omega < 2 \tag{4.48}$$

Exemplo 4.12 *Seja o sistema*

$$\begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix} x = \begin{bmatrix} 1 \\ 3 \\ 3 \\ 1 \end{bmatrix}$$

cuja solução é $(0,1,1,0)^T$. *Utilizando-se o método JOR para resolvê-lo, com* $\omega = 0,65$, *a uma tolerância* $\varepsilon = 10^{-2}$, *obtemos:*

$$\begin{aligned} x_0 &= (0,0000, 0,0000, 0,0000, 0,0000)^T \\ x_1 &= (0,3250, 0,9750, 0,9750, 0,3250)^T \\ x_2 &= (0,0163, 0,8937, 0,8937, 0,0163)^T \\ x_3 &= (0,0349, 0,9921, 0,9921, 0,0349)^T \\ x_4 &= (0,0035, 0,9884, 0,9884, 0,0035)^T \\ x_5 &= (0,0038, 0,9986, 0,9986, 0,0038)^T \end{aligned}$$

ou seja, com cinco iterações, obtemos uma aproximação para a solução dentro da tolerância especificada. O método de Jacobi, se utilizado para resolver o mesmo método, não alcança a solução após 200 *iterações.*

Exemplo 4.13 *Seja o sistema*

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

cuja solução é $x^* = (0,1667, 0,3333, 0,3333, 0,1667)^T$. *Utilizando-se o método SOR para resolvê-lo, com* $\omega = 1,1$, *a uma tolerância* $\varepsilon = 10^{-2}$, *obtemos:*

$$\begin{aligned} x_0 &= (0,0000, 0,2750, 0,2750, 0,1513)^T \\ x_1 &= (0,1513, 0,3307, 0,3307, 0,1668)^T \\ x_2 &= (0,1668, 0,3336, 0,3336, 0,1668)^T \end{aligned}$$

ou seja, com três iterações, obtemos uma aproximação para a solução dentro da tolerância especificada (compare com o exemplo 4.8).

4.5 Método do Gradiente

O método do gradiente é indicado para resolver um SELA onde A é uma matriz *simétrica, positivo-definida* (SPD), i.e.

$$x^T Ax > 0, \forall x \in \mathbb{R}^n \tag{4.49}$$

Uma outra característica de matrizes SPD é que todos os seus autovalores são estritamente positivos.

O método baseia-se na relação existente entre a solução de um SELA e a minimização da *forma quadrática*, quando A for SPD. Assim, inicialmente veremos o que é a forma quadrática e alguns exemplos da mesma.

4.5.1 Forma Quadrática

A forma quadrática é uma função vetorial $f : \Re^n \mapsto \Re$ dada por

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \tag{4.50}$$

onde $A \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$ e $c \in \mathbb{R}$.

Por exemplo, considere o sistema

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.51}$$

cuja solução é $x^* = (1,1)^T$. A matriz de coeficientes é SPD e a figura 4.5 mostra o gráfico e as curvas de nível da forma quadrática correspondente (com $c = 0$). Note que o gráfico da função é um parabolóide – portanto, com apenas um ponto de mínimo – e que aparentemente, o ponto $(1,1)$ (ou seja, a solução do sistema) é o ponto de mínimo da função.

Já as figuras 4.6-4.8 mostram outras situações possíveis, dependendo dos valores dos elementos de A (todos os sistemas tiveram fixada a sua solução em $(1,1)$ e os termos independentes foram calculados adequadamente). Por exemplo, na figura 4.6, temos os gráficos para a matriz *negativo-definida*

$$\begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}$$

i.e., $x^T Ax < 0$, $\forall x$; note que a forma do gráfico é um parabolóide invertido, com apenas um ponto de *máximo* – é a situação oposta à de uma matriz SPD.

Na figura 4.7, temos o caso em que a forma quadrática assume tanto valores negativos quanto positivos – o gráfico da função é a chamada *sela*. A matriz em questão é

$$\begin{bmatrix} 1 & -4 \\ -4 & -5 \end{bmatrix}$$

Finalmente, o gráfico e as curvas de nível para a forma quadrática exibidos na figura 4.8 correspondem a uma matriz *quase-singular*,

$$\begin{bmatrix} 1 & 2 \\ 2 & 3,8 \end{bmatrix}$$

a qual apresenta infinitas soluções ao longo da reta na base do gráfico.

Para verificarmos se isso é verdade, vamos calcular $f'(x) = 0$. Como f é uma função vetorial, a sua derivada – ou *gradiente* – é dada por

$$f'(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix} \tag{4.52}$$

o qual representa um *campo vetorial*; para um dado ponto x, ele aponta na direção de maior variação de $f(x)$. O gráfico de $f'(x)$ na figura 4.9 é típico da situação em que A é SPD:

Aplicando a equação (4.52) à (4.50), obtemos

$$f'(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b \tag{4.53}$$

e, se A é simétrica, $A = A^T$, de onde

$$f'(x) = Ax - b.$$

Igualando $f'(x)$ a zero, obtemos $Ax = b$, ou seja, o sistema que queremos resolver. Portanto, podemos dizer que

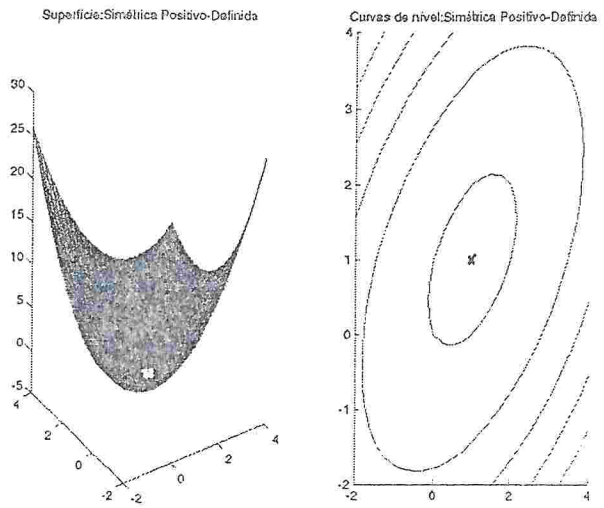

Figura 4.5: *Gráfico de $f(x)$ e suas curvas de nível para A SPD.*

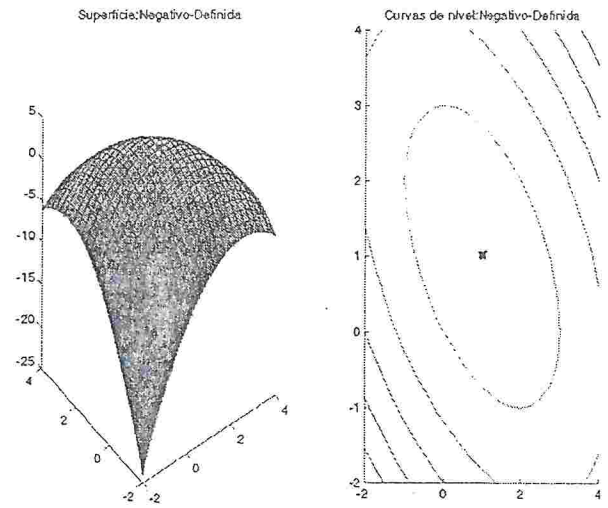

Figura 4.6: *Gráfico de $f(x)$ e suas curvas de nível para A ND.*

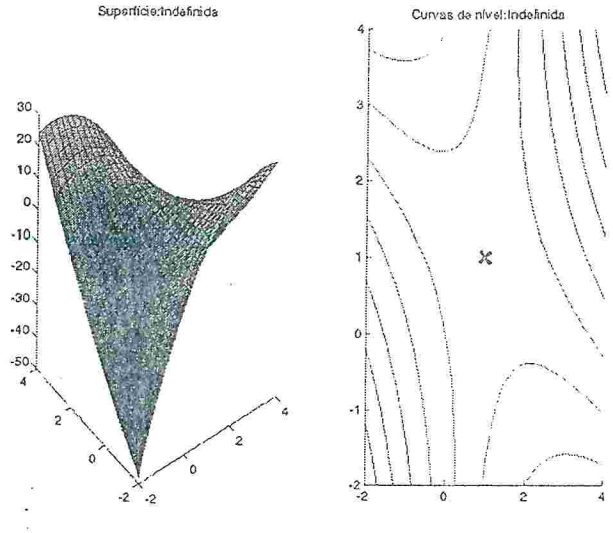

Figura 4.7:. *Gráfico de $f(x)$ e suas curvas de nível para A indefinida.*

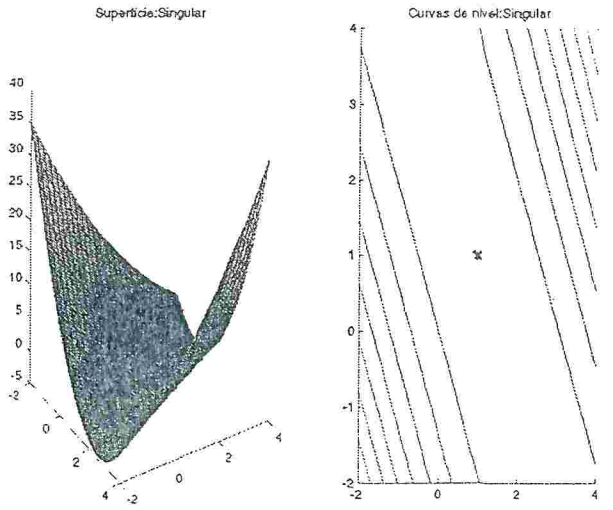

Figura 4.8: *Gráfico de $f(x)$ e suas curvas de nível para A singular.*

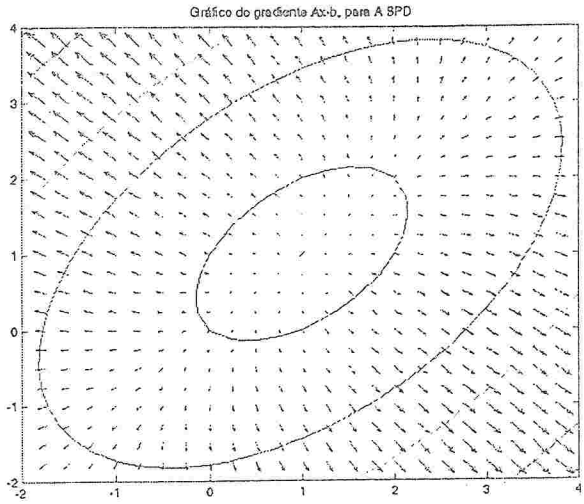

Figura 4.9: *Gráfico de $f'(x)$ para A SPD.*

minimizar a forma quadrática $f(x) = \frac{1}{2}x^TAx - b^Tx + c$ *equivale a resolver o sistema* $Ax = b$ *se* A *for simétrica.*

Se a matriz A é positivo-definida, além de simétrica, então a solução de $Ax = b$ é o mínimo (único) de $f(x)$; logo, para A SPD, a solução $x = A^{-1}b$ é o ponto x que minimiza $f(x)$. Isso pode ser mostrado como segue.

Suponha A simétrica, x um vetor que satisfaz $Ax = b$, y um vetor similar a x (em termos geométricos, y é um ponto próximo a x) e $e = y - x$ o vetor erro; então,

$$\begin{aligned} f(x+e) &= \frac{1}{2}(x+e)^TA(x+e) - b^T(x+e) + c \\ &= \frac{1}{2}x^TAx + e^TAx + \frac{1}{2}e^TAe - b^Tx - b^Te + c \quad \therefore b = Ax; A = A^T \therefore \\ &= \frac{1}{2}x^TAx - b^Tx + c + e^Tb + \frac{1}{2}e^TAe - b^Te \\ &= f(x) + \frac{1}{2}e^TAe \end{aligned} \tag{4.54}$$

e

$$f(x+e) = f(x-x+y) = f(y) = f(x) + \frac{1}{2}(y-x)^TA(y-x) \tag{4.55}$$

Agora, como A é SPD, por hipótese, então

$$(y-x)^TA(y-x) > 0, \quad \forall y$$

e, portanto, $f(y) > f(x)$. Isso mostra que x é o mínimo de $f(x)$, nesse caso.

4.5.2 Descrição do método do Gradiente

O gráfico da forma quadrática, para A SPD, nos sugere uma estratégia para localizarmos a solução do sistema: basta "escorregar" ao longo das paredes do parabolóide, pois isso nos levará, necessariamente, ao ponto de mínimo. A questão que se coloca agora é: qual *direção* devemos tomar, a partir de um x_k, para obtermos um x_{k+1} que seja mais próximo da solução?

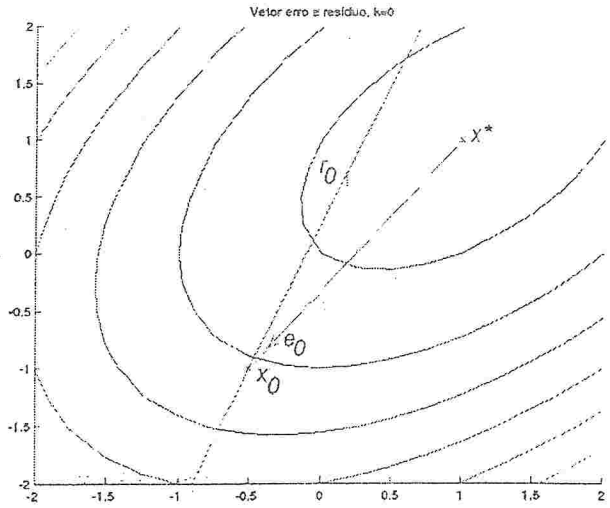

Figura 4.10: *O vetor erro e_0 e o vetor resíduo r_0.*

Lembramos que o gradiente $f'(x)$ aponta na direção de maior aumento de $f(x)$, em sentido *oposto* ao "fundo" do parabolóide. É natural, portanto, que andemos ao longo da direção oposta ao gradiente, isto é, $-f'(x) = b - Ax$. Ora, conforme visto anteriormente, $r_k = b - Ax_k$, de onde estabelecemos as seguintes relações entre o vetor resíduo e o gradiente de $f(x)$:

$$\begin{aligned} r_k &= -f'(x) && (4.56)\\ r_k &= b - Ax_k \therefore e_k = x_k - x^* \therefore \\ &= b - Ax^* - Ae_k \therefore x^* = A^{-1}b \therefore \\ &= -Ae_k && (4.57) \end{aligned}$$

A equação (4.56) nos diz que o resíduo tem a mesma direção do gradiente, porém sentido oposto; já a equação (4.57) nos diz que o resíduo é o vetor erro, transformado por A (e, portanto, no mesmo espaço de b).

Suponha, então, que temos a seguinte situação, conforme a figura 4.10. Como decidimos andar ao longo do vetor resíduo, a partir de x_0, a nova estimativa x_1 é um ponto sobre a reta r_0, ou seja

$$x_1 = x_0 + \alpha_0 r_0, \quad \alpha_0 \in \mathbb{R} \tag{4.58}$$

O escalar α_0 indica o *deslocamento* sobre r_0. Para determinar o melhor α_0 – ou seja, aquele para o qual $\| x_1 - x^* \|$ é mínimo – derivamos $f(x_1)$ em relação a α_0 e igualamo-la a zero:

$$\frac{d}{d\alpha_0} f(x_1) = f'(x_1)^T \frac{d}{d\alpha_0} x_1 = f'(x_1)^T r_0 \tag{4.59}$$

Note que $f'(x_1)^T r_0$ é o produto escalar entre os vetores $f'(x_1)$ e r_0. Como o produto escalar é dado por

$$u^T v = \| u \| \| v \| \cos\theta$$

onde θ é o ângulo formado entre os vetores u e v, ao igualarmos $f'(x_1)^T r_0$ a zero, estamos exigindo que os vetores $f'(x_1)$ e r_0 sejam *ortogonais* entre si. Como $f'(x_1) = -r_1$, isso implica que dois *resíduos* sucessivos são ortogonais entre si; a figura 4.11 mostra três situações típicas para a solução do sistema (4.51), com as seqüências de resíduos (representados pelas retas) gerados pelo método do Gradiente a partir de três diferentes estimativas iniciais.

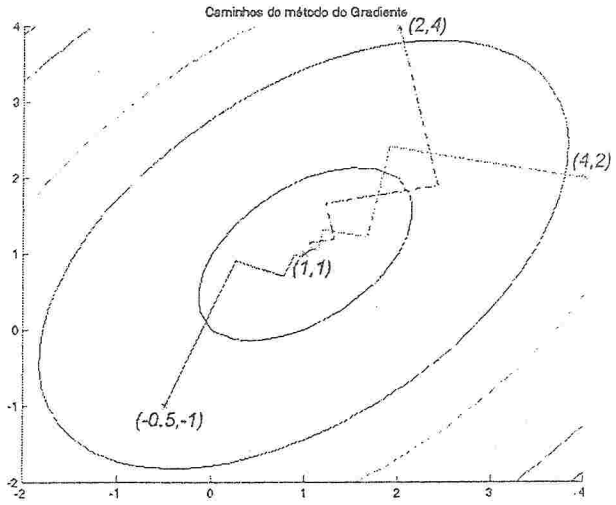

Figura 4.11: *Caminhos típicos no método do Gradiente.*

A partir da equação (4.59), pode-se o obter o valor de α_0:

$$\begin{aligned} f'(x_1)^T r_0 &= 0 \therefore f'(x_1) = -r_1 \therefore \\ -r_1^T r_0 &= 0 \\ (b - Ax_1)^T r_0 &= 0 \\ (b - Ax_0 - \alpha_0 Ar_0)^T r_0 &= 0 \\ (r_0 - \alpha_0 Ar_0)^T r_0 &= 0 \end{aligned}$$

de onde

$$\alpha_0 = \frac{r_0^T r_0}{r_0^T A r_0} \tag{4.60}$$

O que significa minimizar $f'(x_1)$? Os gráficos mostrados na figura 4.12 mostram que, para α_0 calculado conforme a equação (4.60), a nova estimativa x_1 corresponde ao *mínimo* da parábola obtida como se tivéssemos "cortado" o paraboloide $f(x)$ por um plano vertical ao plano $x - y$ que passa pela reta r_0!

Utilizando as equações (4.58) e (4.60), além da expressão para o resíduo, devidamente generalizadas para a k-ésima iteração, podemos escrever um algoritmo que descreve o método do Gradiente. Antes, porém, note que

$$r_1 = r_0 - \alpha_0 Ar_0$$

conforme obtido na derivação da equação (4.60); essa expressão nos permite economizar um produto matriz-vetor da forma Ax_k, pois Ar_k já terá sido calculado previamente para se obter α_k. O algoritmo pode ser, então, escrito como segue.

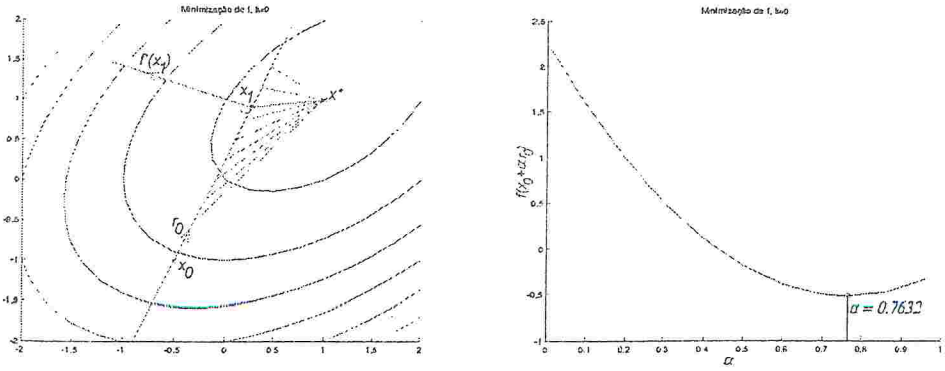

Figura 4.12: α *é escolhido de tal forma que* $f(x_1)$ *é mínima.*

Algoritmo 4.5.1 *Método do Gradiente*

proc gradiente(input: A, b, x_0, $k_{\max}$, ε; *output:* x_{k+1}*)*
  $t \leftarrow \varepsilon \| b \|$
  $r_0 \leftarrow b - Ax_0$
  for $k = 0, 1, \ldots, k_{\max}$ *do*
    $w_k \leftarrow Ar_k$
    $\alpha_k \leftarrow \frac{r_k^T r_k}{r_k^T w_k}$
    $x_{k+1} \leftarrow x_k + \alpha_k r_k$
    $r_{k+1} \leftarrow r_k - \alpha_k w_k$
    if $\| r_{k+1} \| < t$ *then*
      break
    endif
  endfor
endproc

O exemplo seguinte ilustra o comportamento típico do método do Gradiente:

Exemplo 4.14 *Resolva o sistema*

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

cuja solução é $x^* = (0,1667, 0,3333, 0,3333, 0,1667)^T$*, usando o método do Gradiente com* $x_0 = (0,0,0,0)^T$ *a uma tolerância* $\varepsilon = 10^{-2}$.

Solução: *Aplicando o método do Gradiente, obtemos*

$$\begin{aligned} x_1 &= (0,0,25, 0,25, 0)^T \\ x_2 &= (0,125, 0,25, 0,25, 0,125)^T \\ x_3 &= (0,125, 0,3125, 0,3125, 0,125)^T \\ x_4 &= (0,1563, 0,3125, 0,3125, 0,1563)^T \\ x_5 &= (0,1563, 0,3281, 0,3281, 0,1563)^T \\ x_6 &= (0,1641, 0,3281, 0,3281, 0,1641)^T \end{aligned}$$

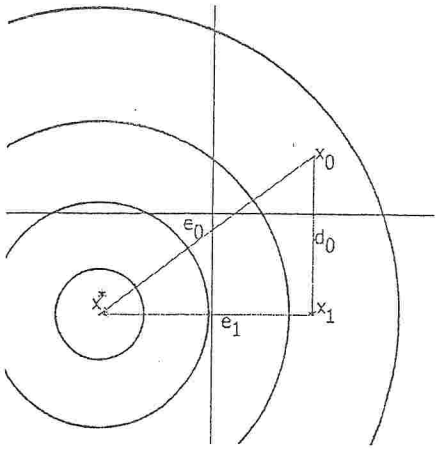

Figura 4.13: *Método das Direções-Conjugadas: a cada iteração é corrigida uma componente do vetor solução.*

ou seja, com seis iterações, obtemos uma aproximação para a solução dentro da tolerância especificada.

Um possível problema ao se utilizar a fórmula de recorrência para r_{k+1} no método do Gradiente é a perda de ortogonalidade entre os vetores resíduo, causada pela acumulação de erros de ponto-flutuante. Isso pode ser detectado através do cálculo do produto-interno entre dois resíduos sucessivos, $r_{k-1}^T r_k$; caso essa quantidade seja maior do que, por exemplo, $\sqrt{\varepsilon}$ (onde ε é o épsilon da máquina), deve-se recalcular $r_k = b - Ax_k$, e proceder normalmente com o algoritmo.

4.6 Método das Direções-Conjugadas

Conforme visto anteriormente, o método do Gradiente toma sucessivas direções – os resíduos – que são ortogonais entre si. Isso significa que a solução é procurada *repetindo-se* direções. Ora, se para uma dada direção, a solução não foi encontrada ao longo dela, então por que utilizá-la novamente?

Uma alternativa é a seguinte: suponha que tenhamos um conjunto de n *direções de procura* $d_0, d_1, \ldots, d_{n-1}$. Em cada i-ésima direção, "caminha-se" exatamente a distância necessária para se obter a i-ésima componente da solução, x_i^*; após termos percorrido todas as n direções dessa forma, todas as componentes de x^* estarão corretas, e a solução terá sido obtida. Na figura 4.13, a primeira iteração corrige x_2, e a segunda corrige x_1 (note que e_1 é ortogonal a d_0).

De forma semelhante ao método do Gradiente, as iterações são da forma

$$x_{i+1} = x_i + \alpha_i d_i, \quad \alpha_i \in \mathbb{R} \tag{4.61}$$

e, para determinar α_i, tomamos e_{i+1} ortogonal a d_i, de forma que não mais se percorra a direção d_i. Então:

$$\begin{aligned} d_i^T e_{i+1} &= 0 \therefore e_{i+1} = x_{i+1} - x^* = x_i + \alpha_i d_i - x^* = e_i + \alpha_i d_i \therefore \\ d_i^T (e_i + \alpha_i d_i) &= 0 \\ d_i^T e_i + \alpha_i d_i^T d_i &= 0 \end{aligned}$$

de onde

$$\alpha_i = \frac{d_i^T e_i}{d_i^T d_i} \tag{4.62}$$

Infelizmente esta equação é inútil, pois para calcular α_i é necessário e_i; porém, se e_i fosse conhecido, a solução seria imediata, por definição.

Podemos corrigir essa situação se considerarmos a determinação de α_i como um problema de minimização ao longo da direção d_i, de maneira análoga ao método do Gradiente. Nesse caso, temos:

$$\begin{aligned} \frac{d}{d\alpha} f(x_{i+1}) &= 0 \\ f'(x_{i+1})^T \frac{d}{d\alpha} x_{i+1} &= 0 \therefore \text{por (4.56) e (4.61), vem} \\ -r_{i+1}^T d_i &= 0 \therefore \text{por (4.57), vem} \end{aligned}$$

$$d_i^T A e_{i+1} = 0 \tag{4.63}$$

A equação (4.63) nos diz que o vetor erro e_{i+1}, transformado para o espaço gerado pelas colunas de A, é ortogonal a d_i. Quaisquer dois vetores u e v que satisfaçam $u^T A v = 0$ são ditos A-ortogonais entre si.

De posse da equação (4.63), podemos determinar uma outra expressão para α_i, a qual pode, dessa vez, ser calculada:

$$\begin{aligned} d_i^T A e_{i+1} &= 0 \therefore e_{i+1} = e_i + \alpha_i d_i \therefore \\ d_i^T A e_i + \alpha_i d_i^T A d_i &= 0 \\ \alpha_i &= -\frac{d_i^T A e_i}{d_i^T A d_i} \therefore \text{por (4.57), vem} \end{aligned}$$

$$\alpha_i = \frac{d_i^T r_i}{d_i^T A d_i} \tag{4.64}$$

Note que, se $d_i \equiv r_i$, então temos a mesma fórmula utilizada para α_i no método do Gradiente.

Conforme proposto quando da motivação do método das Direções-Conjugadas, vejamos como o método converge em n passos. Podemos expressar o vetor erro e_0 como combinação linear dos vetores direção,

$$e_0 = \sum_{j=0}^{n-1} \delta_j d_j \tag{4.65}$$

Para determinarmos as constantes δ_j, valemo-nos da propriedade de A-ortogonalidade entre os vetores d_j; pré-multiplicando (4.65) por $d_k^T A$, obtemos

$$d_k^T A e_0 = \sum_{j=0}^{n-1} \delta_j d_k^T A d_j$$

e, como $d_k^T A d_j = 0$, $\forall i \neq j$, podemos eliminar todos os termos do somatório, menos o termo para $j = k$, de onde

$$\begin{aligned} d_k^T A e_0 &= \delta_k d_k^T A d_k \\ \delta_k &= \frac{d_k^T A e_0}{d_k^T A d_k} \end{aligned} \tag{4.66}$$

Precisamos, ainda, encontrar uma equação para e_0 que não envolva δ_k; escrevendo as expressões para os vetores erro, temos

$$\begin{aligned} e_0 &= x_0 - x^* \\ e_1 &= x_1 - x^* = e_0 + \alpha_0 d_0 = x_0 - x + \alpha_0 d_0 = e_0 + \sum_{i=0}^{0} \alpha_i d_i \end{aligned}$$

$$
\begin{aligned}
e_2 &= x_2 - x^* = e_1 + \alpha_1 d_1 = x_0 - x + \alpha_1 d_1 + \alpha_0 d_0 = e_0 + \sum_{i=0}^{1} \alpha_i d_i \\
&\vdots \quad \vdots \quad \ddots \\
e_k &= x_k - x^* = e_{k-1} + \alpha_{k-1} d_{k-1} = x_0 - x + \alpha_{k-1} d_{k-1} + \ldots + \alpha_0 d_0 = e_0 + \sum_{i=0}^{k-1} \alpha_i d_i
\end{aligned}
$$

de onde

$$e_0 = e_k - \sum_{i=0}^{k-1} \alpha_i d_i \tag{4.67}$$

Agora, substituindo a equação (4.67) em (4.66), obtemos

$$\delta_k = \frac{d_k^T A e_k - \sum_{i=0}^{k-1} \alpha_i d_k^T A d_i}{d_k^T A d_k}$$

e, como $d_k^T A d_i = 0, \forall k \neq i$, o segundo termo no numerador é nulo (pois apenas os vetores d_0, d_1, ..., d_{k-1} aparecem no somatório). Assim, obtemos

$$\delta_k = \frac{d_k^T A e_k}{d_k^T A d_k} \tag{4.68}$$

Comparando as equações (4.68) e (4.64), vemos que $\alpha_i = -\delta_i$. Podemos, então, reescrever (4.67) como

$$
\begin{aligned}
e_i &= e_0 - \sum_{j=0}^{k-1} \alpha_j d_j \\
&= \sum_{j=0}^{n-1} \alpha_j d_j - \sum_{j=0}^{k-1} \alpha_j d_j \\
&= \sum_{j=k}^{n-1} \alpha_j d_j
\end{aligned} \tag{4.69}
$$

A equação (4.69) pode ser interpretada da seguinte forma: quando $k = 0$, i.e. na primeira iteração, todas as componentes de x^* estão erradas (em princípio), logo e_0 é combinação linear de todas as direções de busca d_i. Na segunda iteração, uma componente já foi corrigida – ao longo da direção d_0 – e, portanto, o erro e_1 só deve ter componentes diferentes de zero ao longo das direções d_1, d_2, ..., d_{k-1}. Procedendo com esse raciocínio até e_{n-1}, vemos que o processo de se obter uma componente correta de x^* a cada iteração equivale a se eliminar a componente correspondente do erro a cada iteração.

Neste estágio, resta-nos determinar as direções d_i, de maneira que sejam A-ortogonais entre si. Podemos obter tais direções se tomarmos um conjunto de vetores linearmente independentes u_i, e os A-ortogonalizarmos através de uma modificação do *processo de Gram-Schmidt*. Para se gerar um vetor d_i, subtraem-se de u_i todas as componentes que não sejam A-ortogonais aos $i-1$ vetores d anteriores, ou seja,

$$
\begin{aligned}
d_0 &= u_0 \\
d_i &= u_i + \textstyle\sum_{k=0}^{i-1} \beta_{ik} d_k, \quad i > 0
\end{aligned} \tag{4.70}
$$

Para determinarmos os valores das constantes β_{ik}, pós-multiplicamos a expressão para d_i em (4.70) por Ad_j, de onde

$$
\begin{aligned}
d_i^T A d_j &= u_i^T A d_j + \sum_{k=0}^{i-1} \beta_{ik} d_k^T A d_j; \; |i > j \\
0 &= u_i^T A d_j + \beta_{ij} d_j^T A d_j
\end{aligned}
$$

pela A-ortogonalidade entre os vetores d_i. Assim, podemos escrever

$$\beta_{ij} = -\frac{u_i^T A d_j}{d_j^T A d_j} \tag{4.71}$$

O processo acima requer o armazenamento de todos os n vetores d_i, o que pode não ser desejável para sistemas lineares com n grande.

Cabe notar que, se os vetores u_i são os vetores canônicos (i.e. a i-ésima componente de u_i é igual a 1 e todas as demais são nulas), então o método das Direções-Conjugadas reduz-se à Eliminação Gaussiana.

Um algoritmo que expressa o método das Direções-Conjugadas, pode ser escrito como segue; cabe ressaltar que, apesar do método ser considerado um método direto, por convergir em exatamente n iterações, para um sistema de n equações lineares, ele pode também ser considerado um método iterativo, pois é possível que, devido a erros de arredondamento, a solução seja alcançada em menos do que n iterações.

Algoritmo 4.6.1 *A-ortogonalização*

proc *A_ortogonaliza(input:* U; *output:* D*)*
  % U *e* D *são matrizes* $n \times n$ *cujas colunas*
  % *são os vetores* u_i *e* d_i*, respectivamente.*
  $d_0 \leftarrow u_0$
  for $i \leftarrow 1, 2, \ldots, n$ *do*
    $s \leftarrow 0$
    for $k = 1, 2, \ldots, i-1$ *do*
    $w \leftarrow Ad_k$
    $\beta \leftarrow \frac{u_i^T w}{d_k^T w}$
    $s \leftarrow s - \beta d_k$
    endfor
    $d_i \leftarrow d_i + s$
  endfor
endproc

Algoritmo 4.6.2 *Método das Direções-Conjugadas*

proc *direções_conjugadas(input:* A, b, U, x_0, $k_{\max}$, ε; *output:* x_{k+1}*)*
  call *A_ortogonaliza(U;D)*
  $t \leftarrow \varepsilon || b ||$
  $r_0 \leftarrow b - Ax_0$
  for $k = 0, 1, \ldots, k_{\max}$ *do*
    $w_k \leftarrow Ad_k$
    $\alpha_k \leftarrow \frac{d_k^T r_k}{d_k^T w_k}$
    $x_{k+1} \leftarrow x_k + \alpha_k d_k$
    $r_{k+1} \leftarrow r_k - \alpha_k w_k$
    if $|| r_{k+1} || < t$ *then*
      break
    endif
  endfor
endproc

O exemplo a seguir ilustra o comportamento típico do método das Direções-Conjugadas:

Exemplo 4.15 *Resolva o sistema*

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

cuja solução é $x^* = (0,1667, 0,3333, 0,3333, 0,1667)^T$, *usando o método das Direções-Conjugadas com* $x_0 = (0,0,0,0)^T$ *a uma tolerância* $\varepsilon = 10^{-2}$; *os vetores* u_i *são tomados como vetores aleatórios.*

Solução: *Aplicando o método das Direções-Conjugadas, obtemos*

$$\begin{aligned} x_1 &= (0,2113, 0,2071, 0,0927, 0,2018)^T \\ x_2 &= (0,1950, 0,2941, 0,3595, 0,1705)^T \\ x_3 &= (0,1986, 0,2961, 0,3578, 0,1649)^T \\ x_4 &= (0,1667, 0,3333, 0,3333, 0,1667)^T \end{aligned}$$

ou seja, com quatro iterações, obtemos uma aproximação para a solução, com um resíduo da ordem de 10^{-15}. *Normalmente, no caso desse método, o resíduo da solução obtida será um número próximo ao épsilon da máquina.*

4.7 Método dos Gradientes-Conjugados

O método dos Gradientes-Conjugados nada mais é do que o método das Direções-Conjugadas, onde os vetores u_i são tomados como os vetores resíduo r_i. Essa escolha nos permitirá simplificar sobremaneira o processo de Gram-Schmidt descrito anteriormente.

Suponha o espaço gerado pelos vetores direção. Pela equação (4.67), vemos que o i-ésimo vetor e_i é gerado como combinação linear dos vetores d_i e pelo vetor erro e_0. Podemos, então, derivar algumas importantes relações que serão utilizadas a seguir. Se pré-multiplicarmos a equação (4.69) por $-d_i^T A$, vem

$$-d_i^T A e_j = \sum_{j=i}^{n-1} \alpha_j d_i^T A d_j$$

e, pela A-ortogonalidade entre os vetores d_i, temos

$$d_i^T r_j = 0, \quad i < j \tag{4.72}$$

Além disso, o i-ésimo resíduo é ortogonal aos $i-1$ vetores u,

$$d_i^T r_j = u_i^T r_j + \sum_{k=0}^{i-1} \beta_{ik} d_k^T r_j \tag{4.73}$$

$$0 = u_i^T r_j, \quad i < j \tag{4.74}$$

de onde

$$d_i^T r_i = u_i^T r_i \tag{4.75}$$

Como os vetores u_i são tomados como os vetores resíduo r_i, a equação (4.74) pode ser reescrita como

$$r_i^T r_j = 0, \quad i \neq j \tag{4.76}$$

De maneira semelhante ao utilizado no método do Gradiente, podemos obter uma fórmula de recorrência para r_{i+1},

$$\begin{aligned} r_{i+1} &= -Ae_{i+1} = -A(e_i + \alpha_i d_i) = \\ &= -Ae_i - \alpha_i A d_i \end{aligned}$$

de onde

$$r_{i+1} = r_i - \alpha_i A d_i \tag{4.77}$$

Agora, de posse das equações (4.76) e (4.77), podemos simplificar a expressão para β_{ij}. Substituindo i por j em (4.77) e pré-multiplicando-a por r_i, vem

$$\begin{aligned} r_i^T r_{j+1} &= r_i^T r_j - \alpha_j r_i^T A d_j \\ \alpha_j r_i^T A d_j &= r_i^T r_j - r_i^T r_{j+1} \end{aligned}$$

de onde

$$r_i^T A d_j = \begin{cases} \frac{1}{\alpha_i} r_i^T r_i, & i = j \\ -\frac{1}{\alpha_{i-1}} r_i^T r_i, & i = j+1, \quad \text{pela equação (4.76)} \\ 0, & \text{c.c.} \end{cases} \tag{4.78}$$

e, então, β_{ij} pode ser escrito como

$$\beta_{ij} = \begin{cases} \frac{1}{\alpha_{i-1}} \frac{r_i^T r_i}{d_{i-1}^T A d_{i-1}}, & i = j+1 \\ 0, & i > j+1 \end{cases} \tag{4.79}$$

Devido à A-ortogonalidade entre os vetores direção e os resíduos, a grande maioria dos termos necessários à formulação de β_{ij} pode ser descartada. Utilizando as equações (4.64), (4.75) e (4.76), a equação (4.79) pode ser simplificada ainda mais:

$$\begin{aligned} \beta_i &= \frac{d_{i-1}^T A d_{i-1}}{d_{i-1}^T r_{i-1}} \frac{r_i^T r_i}{d_{i-1}^T A d_{i-1}} = \frac{r_i^T r_i}{d_{i-1}^T r_{i-1}} = \\ &= \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}} \end{aligned} \tag{4.80}$$

onde o subscrito em j foi descartado, por ser desnecessário.

Tendo obtido essa expressão para β_i, podemos combiná-la com as equações (4.61), (4.64) e (4.77), de forma a escrever um algoritmo que expressa o método dos Gradientes-Conjugados, conforme ilustrado a seguir. Note que, a partir da equação (4.70), e considerando que $u_i \equiv r_i$, expressamos um vetor direção como

$$d_{i+1} = r_{i+1} + \beta_{i+1} d_i \tag{4.81}$$

Algoritmo 4.7.1 *Gradientes-Conjugados*

```
proc gradientes_conjugados(input: A, b, x_0, k_max, ε; output: x_{k+1})
    t ← ε|| b ||
    r_0 ← b − A x_0
    d_0 ← r_0
    for k = 0, 1, ..., k_max do
        w_k ← A d_k
        α_k ← (r_k^T r_k) / (d_k^T w_k)
        x_{k+1} ← x_k + α_k d_k
        r_{k+1} ← r_k − α_k w_k
        if || r_{k+1} || < t then
            break
        endif
        β_{k+1} ← (r_{k+1}^T r_{k+1}) / (r_k^T r_k)
        d_{k+1} ← r_{k+1} + β_{k+1} d_k
    endfor
endproc
```

O exemplo seguinte ilustra o comportamento típico do método dos Gradientes-Conjugados:

Exemplo 4.16 *Resolva o sistema*

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

cuja solução é $x^* = (0,1667, 0,3333, 0,3333, 0,1667)^T$, *usando o método dos Gradientes-Conjugados com* $x_0 = (0,0,0,0)^T$ *a uma tolerância* $\varepsilon = 10^{-2}$.

Solução: *Aplicando o método dos Gradientes-Conjugados, obtemos*

$$\begin{aligned} x_1 &= (0,0,25, 0,25, 0)^T \\ x_2 &= (0,1667, 0,3333, 0,3333, 0,1667)^T \end{aligned}$$

ou seja, com apenas duas iterações, obtemos uma aproximação para a solução dentro da tolerância especificada.

4.8 Exercícios

Exercício 4.1 *Resolva o sistema*

$$\begin{cases} 4x_1 + 4x_2 = 20,5 \\ 7x_1 + 6,99x_2 = 34,97 \end{cases}$$

através do método de eliminação de Gauss, com precisão de 5 dígitos significativos, refinando a solução obtida.

Exercício 4.2 *Calcule* norm$|A|$ *das matrizes*

$$(a) \begin{bmatrix} 0,992 & 0,873 \\ 0,481 & 0,421 \end{bmatrix}, \quad (b) \begin{bmatrix} 1 & 5 \\ 1,5 & 7,501 \end{bmatrix}$$

e diga se são bem ou mal-condicionadas.

Exercício 4.3 *Calcule o número de condição das matrizes do exercício 4.2, sabendo que as suas inversas são dadas de forma aproximada, respectivamente, por*

$$(a) \begin{bmatrix} -184,568 & 382,727 \\ 210,872 & -434,897 \end{bmatrix}, \quad (b) \begin{bmatrix} 7501 & -5000 \\ -1500 & 1000 \end{bmatrix}$$

Exercício 4.4 *Resolva o sistema*

$$\begin{cases} 10x_1 + x_2 + x_3 = 12 \\ x_1 + 10x_2 + x_3 = 12 \\ x_1 + x_2 + 10x_3 = 12 \end{cases}$$

através do método de Jacobi, com uma tolerância de 10^{-7}.

Exercício 4.5 *Resolva o sistema do exercício 4.4 através do método de Gauss-Seidel, com uma tolerância de* 10^{-7}.

Exercício 4.6 *Resolva o sistema do exercício 4.4 através do método SOR, com uma tolerância de* 10^{-7}. *Determine, experimentando diversos valores, um* ω *que reduza substancialmente o número de iterações necessárias à convergência.*

Exercício 4.7 *Verifique que o sistema linear*

$$\begin{bmatrix} 3 & 0 & 1 \\ 1 & -1 & 0 \\ 3 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

não é diagonal-dominante, apesar do critério de Sassenfeld ser satisfeito.

Exercício 4.8 *Explique o que acontece com a aplicação do método de Gauss-Seidel ao sistema*

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 5 & 5 & 5 \end{bmatrix} x = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}.$$

Exercício 4.9 *Utilize o método do Gradiente e resolva o sistema do exercício 4.8; explique o que ocorre.*

Exercício 4.10 *Utilize o método dos Gradientes-Conjugados e resolva o sistema do exercício 4.8; explique o que ocorre.*

Exercício 4.11 *Mostre, utilizando o sistema*

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} x = \begin{bmatrix} -10 \\ 10 \end{bmatrix}$$

que o método das Direções-Conjugadas equivale ao método da eliminação Gaussiana, se as direções tomadas são os vetores $u_1 = (1,0)^T$ *e* $u_2 = (0,1)^T$.

Exercício 4.12 *Resolva o sistema*

$$\begin{bmatrix} 10^{-6} & -1 \\ 1 & 1 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

através do método dos Gradientes-Conjugados, para uma tolerância de 10^{-4}.

Capítulo 5

Resolução de Sistemas de Equações Não-Lineares

5.1 Introdução

Neste capítulo, apresentaremos o método de Newton para sistemas de equações *não-lineares*, i.e., procuramos um vetor x que satisfaça

$$F(x) = 0 \tag{5.1}$$

onde $F : \mathbb{R}^n \to \mathbb{R}^n$, i.e.

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases} \tag{5.2}$$

Pode-se afirmar que todas as considerações apresentadas no Capítulo 2 para o método de Newton-Raphson são também válidas para esse caso. No entanto, a solução de (5.1) é bem mais difícil, requerendo uma série de cuidados adicionais (ver [5]).

5.2 Método de Newton

Como visto na seção 2.4, o método de Newton-Raphson é uma linearização da função $f(x)$ no ponto $x = x_k$. Essa idéia deve ser estendida para o presente caso, como veremos a seguir.

Considerando então o sistema (5.1), podemos escrever as expansões em Taylor (apenas até os termos de primeira ordem) de cada função f_i em (5.2) como

$$\begin{cases} 0 = f_1(x_1 + h_1, x_2 + h_2, \ldots, x_n + h_n) \approx f_1(x_1, x_2, \ldots, x_n) + h_1 \frac{\partial f_1}{\partial x_1} + h_2 \frac{\partial f_1}{\partial x_2} + \ldots + h_n \frac{\partial f_1}{\partial x_n} \\ \vdots \\ 0 = f_n(x_1 + h_1, x_2 + h_2, \ldots, x_n + h_n) \approx f_n(x_1, x_2, \ldots, x_n) + h_1 \frac{\partial f_n}{\partial x_1} + h_2 \frac{\partial f_n}{\partial x_n} + \ldots + h_n \frac{\partial f_n}{\partial x_n} \end{cases} \tag{5.3}$$

ou, em termos matriciais:

$$\begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.4}$$

$$F(x) + J(x)h = 0$$

onde $J(x)$ é a matriz *Jacobiana* de $F(x)$. Ora, para obtermos o vetor $h = (h_1, h_2, \ldots, h_n)$, devemos resolver o sistema de equações *lineares*

$$J(x)h = -F(x) \tag{5.5}$$

o que exige, obviamente, que $J(x)$ seja não-singular. Então, se $x \equiv x^{(k)}$ (i.e., x é uma estimativa para a solução de (5.1) na iteração k), podemos obter uma nova estimativa (possivelmente melhor) através de

$$\begin{bmatrix} x_1^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} + \begin{bmatrix} h_1^{(k)} \\ \vdots \\ h_n^{(k)} \end{bmatrix} \tag{5.6}$$

$$x^{(k+1)} = x^{(k)} + h^{(k)}$$

Note as similaridades com o método de Newton-Raphson: naquele, a correção é escrita como $x_{k+1} = x_k + h_k$, onde $h_k = f(x)/f'(x)$, o que é equivalente à equação (5.5).

Ao se resolver o sistema (5.5), pode-se utilizar qualquer um dos métodos vistos no Capítulo 4; usualmente, utiliza-se a fatoração LU (vide seção 4.3.2), mas métodos iterativos são indicados quando a matriz J é esparsa [9]. De qualquer maneira, no entanto, a matriz J pode se tornar quase singular, o que dificulta bastante a solução de (5.1).

Outro problema relacionado ao método de Newton é na obtenção da matriz $J(x)$. Cabe notar que apenas para problemas com n muito pequeno é factível calcular-se de forma explícita a matriz J; logo, J deve ser calculada de forma aproximada. Ora, seus elementos são as derivadas parciais das f_i em relação a x_i; portanto, aproximações dessas derivadas por diferenças finitas (vide 2.5) podem ser utilizadas. A definição típica de cada coluna da matriz $J(x)$ é

$$J(x)_j = \begin{cases} \frac{F(x+h\|x\|e_j)-F(x)}{h\|x\|} & x \neq 0 \\ \frac{F(he_j)-F(x)}{h} & x = 0 \end{cases} \tag{5.7}$$

conforme [9, pp. 80], onde e_j é o vetor canônico de n elementos, $h = \sqrt{\varepsilon}$ e ε é o épsilon da máquina (vide Capítulo 1).

O método de Newton para sistemas de equações não-lineares pode, então, ser descrito conforme o algoritmo 5.2.1, onde τ_r e τ_a são duas tolerâncias pré-especificadas.

Algoritmo 5.2.1 *Método de Newton para sistemas de equações não-lineares*

proc newton(input: x_0, τ_r, τ_a, $k_{\max}$; *output:* x_{k+1}, k)
  $F_0 = \|F(x_0)\|$
  for $k = 0, 1, \ldots, k_{\max}$ *do*
    Calcule $J(x^{(k)})$
    Resolva o sistema $J(x^{(k)})h^{(k)} = -F(x^{(k)})$
    $x^{(k+1)} \leftarrow x^{(k)} + h^{(k)}$
    if $\|F(x^{(k)})\| < \tau_r F_0 + \tau_a$ *then*
      break
    endif
    Calcule $F(x^{(k+1)})$
  endfor
endproc

Os exemplo a seguir ilustram como utilizar o método de Newton.

Exemplo 5.1 *Sejam as equações*

$$\begin{aligned} f(x,y) &= \text{sen}(y)e^{-x} - x^2 \\ g(x,y) &= \cos(y)e^{-x} - x^3 \end{aligned}$$

Calcule o ponto de intersecção entre ambas as curvas usando o método de Newton.

Solução: O gráfico das funções f e g é mostrado na figura 5.1; nele pode-se perceber que a solução é, aproximadamente, $(0,7, 1, 0)$.

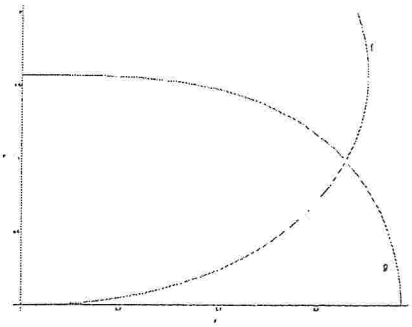

Figura 5.1: *As funções f e g.*

Escrevendo na notação apropriada, temos

$$F = \begin{bmatrix} \text{sen}(y)e^{-x} - x^2 \\ \cos(y)e^{-x} - x^3 \end{bmatrix}$$

e a matriz Jacobiana é dada, explicitamente, por

$$J = \begin{bmatrix} -\text{sen}(y)e^{-x} - 2x & \cos(y)e^{-x} \\ -\cos(y)e^{-x} - 3x^2 & -\text{sen}(y)e^{-x} \end{bmatrix}$$

Assim, utilizando como estimativa inicial o vetor $x_0 = (x, y) = (0, 0)^T$, *obtemos a seguinte seqüência de valores para o método de Newton, conforme o algoritmo 5.2.1:*

$$\begin{aligned}
F_0 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
J_0 &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
h_0 &= -J_0^{-1}F_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
x_1 &= x_0 + h_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
F_1 &= \begin{bmatrix} -1 \\ -0,6321 \end{bmatrix} \\
J_1 &= \begin{bmatrix} -2 & -0,3679 \\ -3,3679 & 0 \end{bmatrix} \\
h_1 &= -J_1^{-1}F_1 = \begin{bmatrix} -0,1877 \\ 1,6979 \end{bmatrix} \\
x_2 &= x_1 + h_1 = \begin{bmatrix} 0,8123 \\ 1,6979 \end{bmatrix} \\
F_2 &= \begin{bmatrix} -0,2196 \\ -0,5923 \end{bmatrix} \\
J_2 &= \begin{bmatrix} -2,0649 & -0,0563 \\ -1,9233 & -0,4403 \end{bmatrix} \\
h_2 &= -J_2^{-1}F_2 = \begin{bmatrix} -0,0791 \\ -0,9996 \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
x_3 &= x_2 + h_2 = \begin{bmatrix} 0,7332 \\ 0,6982 \end{bmatrix} \\
F_3 &= \begin{bmatrix} -0,2288 \\ -0,0262 \end{bmatrix} \\
J_3 &= \begin{bmatrix} -1,7752 & 0,3680 \\ -1,9807 & -0,3088 \end{bmatrix} \\
h_3 &= -J_3^{-1}F_3 = \begin{bmatrix} -0,0629 \\ 0,3184 \end{bmatrix} \\
x_4 &= x_3 + h_3 = \begin{bmatrix} 0,6709 \\ 1,0166 \end{bmatrix} \\
F_4 &= \begin{bmatrix} -0,0144 \\ -0,0320 \end{bmatrix} \\
J_4 &= \begin{bmatrix} -1,7756 & 0,2692 \\ -1,6172 & -0,4350 \end{bmatrix} \\
h_4 &= -J_4^{-1}F_4 = \begin{bmatrix} -0,0123 \\ -0,0279 \end{bmatrix} \\
x_5 &= x_4 + h_4 = \begin{bmatrix} 0,6588 \\ 0,9888 \end{bmatrix} \\
F_5 &= 10^{-3}\begin{bmatrix} -0,3810 \\ -0,2395 \end{bmatrix} \\
J_5 &= \begin{bmatrix} -1,7487 & 0,2847 \\ -1,5837 & -0,4926 \end{bmatrix} \\
h_5 &= -J_5^{-1}F_5 = 10^{-3}\begin{bmatrix} -0,1930 \\ 0,1529 \end{bmatrix} \\
x_6 &= x_5 + h_5 = \begin{bmatrix} 0,6578 \\ 0,9889 \end{bmatrix} \\
F_6 &= 10^{-7}\begin{bmatrix} -0,2585 \\ -0,8432 \end{bmatrix}
\end{aligned}
$$

ou seja, após 6 iterações, o valor de $F(x_6)$ é considerado pequeno o suficiente, e $x = 0,6578$ e $y = 0,9889$ – bastante próximos da estimativa para a solução conforme o gráfico na figura 5.1. Obviamente, poderíamos ter acelerado consideravelmente o processo utilizando como estimativa inicial o vetor $x_0 = (0,7,1)$.

O mesmo número de iterações é alcançado se utilizarmos a aproximação numérica da matriz Jacobiana dada pela equação (5.7).

Exemplo 5.2 *Considere o problema de intersecção de uma reta que passa pelos pontos q_0 e q_1 em $\mathbb{R}^3$,*

$$\vec{r}(t) = q_0 + (q_1 - q_0)t, \quad t \in \mathbb{R}$$

e o plano que passa pelos pontos p_0, p_1 e p_2,

$$\vec{S}(u,v) = (p_0 + (p_1 - p_0)u + (p_2 - p_0)v, \quad u,v \in \mathbb{R}$$

i.e., queremos resolver o problema

$$\vec{S}(u,v) - \vec{r}(t) = 0$$

Se $q_0 = (5,5,0)$, $q_1 = (5,-5,0)$, $p_0 = (0,0,0)$, $p_1 = (0,0,-10)$ e $p_2 = (10,0,0)$, mostre como se comporta o método de Newton nesse caso.

Solução: Note que $\vec{S}(u,v)$ e $\vec{r}(t)$ são funções vetoriais em $\mathbb{R}^3$. Como temos três variáveis a determinar – u, v e t – e três equações da forma $\vec{S}(u,v) - \vec{r}(t) = 0$, uma para cada componente

x, y e z, o problema é bem posto. Como $\vec{S}(u,v)$ é uma função linear, o método de Newton deverá convergir em uma única iteração[1].

Escrevendo então na notação adequada, temos

$$F = \begin{bmatrix} (au + bv - ct + p_0 - q_0)_x \\ (au + bv - ct + p_0 - q_0)_y \\ (au + bv - ct + p_0 - q_0)_z \end{bmatrix}$$

onde $a = p_1 - p_0$, $b = p_2 - p_0$ e $c = q_1 - q_0$, temos que a matriz Jacobiana de F é constante,

$$J = \begin{bmatrix} a_x & b_x & -c_x \\ a_y & b_y & -c_y \\ a_z & b_z & -c_z \end{bmatrix}$$

e, para os valores fixados, temos

$$J = \begin{bmatrix} 0 & 10 & 0 \\ 0 & 0 & 10 \\ -10 & 0 & 0 \end{bmatrix}$$

a qual apresenta inversa e, portanto, o método de Newton não irá sofrer interrupção. Com efeito, se $x_0 = (u,v,t) = (0,0,0)^T$, teremos a seguinte seqüência de valores

$$\begin{aligned} F_0 &= \begin{bmatrix} -5 \\ -5 \\ 0 \end{bmatrix} \\ J_0 &= \begin{bmatrix} 0 & 10 & 0 \\ 0 & 0 & 10 \\ -10 & 0 & 0 \end{bmatrix} \\ h_0 &= -J_0^{-1} F_0 = \begin{bmatrix} 0 \\ 0,5 \\ 0,5 \end{bmatrix} \\ x_1 &= x_0 + h_0 = \begin{bmatrix} 0 \\ 0,5 \\ 0,5 \end{bmatrix} \\ F_1 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

como era esperado, dada a natureza linear do problema. Cabe ressaltar, no entanto, que uma formulação semelhante deve ser usada para problemas onde $\vec{S}(u,v)$ é não-linear.

Exemplo 5.3 *Uma* superfície bicúbica de Bézier $\vec{B}_3(u,v) \in \mathbb{R}^3$ *é definida por*

$$\vec{B}_3(u,v) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} N \vec{P} N^T \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}, \quad , 0 \le u, v \le 1$$

onde N é uma matriz dada por

$$\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

[1] Na verdade, não se utilizaria tal método para se resolver esse problema; ele é útil apenas para fins de ilustração.

e $\overrightarrow{P} \in \mathbb{R}^3$ *é uma matriz que contém* 16 pontos de controle *que governam o desenvolvimento da superfície.*

Um ponto da superfície é dado por três polinômios bicúbicos em u *e* v *tais que, se os pontos de controle são ordenados como*

$$\begin{bmatrix} \overrightarrow{P}_{0,0} & \overrightarrow{P}_{0,1} & \overrightarrow{P}_{0,2} & \overrightarrow{P}_{0,3} \\ \overrightarrow{P}_{1,0} & \overrightarrow{P}_{1,1} & \overrightarrow{P}_{1,2} & \overrightarrow{P}_{1,3} \\ \overrightarrow{P}_{2,0} & \overrightarrow{P}_{2,1} & \overrightarrow{P}_{2,2} & \overrightarrow{P}_{2,3} \\ \overrightarrow{P}_{3,0} & \overrightarrow{P}_{3,1} & \overrightarrow{P}_{3,2} & \overrightarrow{P}_{3,3} \end{bmatrix}$$

então, para $u = 0$ *e* $v = 0$, $\overrightarrow{B}_3(0,0) = P_{0,0}$; *para* $u = 0$ *e* $v = 1$, $\overrightarrow{B}_3(0,1) = P_{0,3}$; $\overrightarrow{B}_3(1,0) = P_{3,0}$ *e* $\overrightarrow{B}_3(1,1) = P_{3,3}$.

Considere, então, o problema de intersecção de um segmento de reta com origem no ponto $(100,5,5)$ *e vetor direção* $(-1,0,0)$,

$$\overrightarrow{r}(t) = (100,5,5) + t(-1,0,0), \quad t \in \mathbb{R}$$

com a superfície $\overrightarrow{B}_3(u,v)$, *onde os pontos de controle são*

$$\overrightarrow{P}_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 \\ 20 & 20 & 20 & 20 \\ 30 & 30 & 30 & 30 \end{bmatrix}$$

$$\overrightarrow{P}_y = \begin{bmatrix} 0 & 5 & 5 & 0 \\ 5 & 10 & 10 & 5 \\ 5 & 10 & 10 & 5 \\ 0 & 5 & 5 & 0 \end{bmatrix}$$

$$\overrightarrow{P}_z = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 \\ 20 & 20 & 20 & 20 \\ 30 & 30 & 30 & 30 \end{bmatrix}$$

de onde a função F *pode ser escrita como*

$$F = \begin{bmatrix} (B_3(u,v))_x + t - 100 \\ (B_3(u,v))_y - 5 \\ (B_3(u,v))_z - 5 \end{bmatrix}.$$

Note que, agora, a matriz Jacobiana não é constante. A figura 5.2 mostra a superfície e a reta em questão.

Se utilizarmos o método de Newton com $x_0 = (u,v,t) = (0,0,0)^T$ *e uma tolerância de* 10^{-10}, *teremos a seguinte seqüência de valores*

$$F_0 = \begin{bmatrix} -100 \\ -5 \\ -5 \end{bmatrix}$$

$$J_0 = \begin{bmatrix} 30 & 0 & 1 \\ 15 & 15 & 0 \\ 30 & 0 & 0 \end{bmatrix}$$

$$h_0 = -J_0^{-1}F_0 = \begin{bmatrix} 0,1667 \\ 0,1667 \\ 95 \end{bmatrix}$$

$$x_1 = x_0 + h_0 = \begin{bmatrix} 0,1667 \\ 0,1667 \\ 95 \end{bmatrix}$$

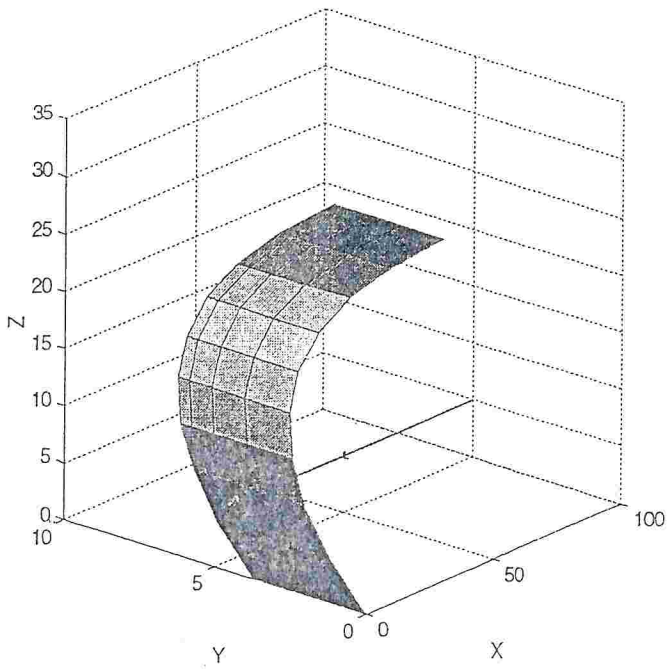

Figura 5.2: *Intersecção entre a reta* $(100, 5, 5) + t(-1, 0, 0)$ *com uma superfície bicúbica de Bézier.*

$$
\begin{aligned}
F_1 &= \begin{bmatrix} 0 \\ -0,8333 \\ 0 \end{bmatrix} \\
J_1 &= \begin{bmatrix} 30 & 0 & 1 \\ 10 & 10 & 0 \\ 30 & 0 & 0 \end{bmatrix} \\
h_1 &= -J_1^{-1}F_1 = \begin{bmatrix} 0 \\ 0,0833 \\ 0 \end{bmatrix} \\
x_2 &= x_0 + h_0 = \begin{bmatrix} 0,1667 \\ 0,2500 \\ 95 \end{bmatrix} \\
F_2 &= \begin{bmatrix} 0 \\ -0,1042 \\ 0 \end{bmatrix} \\
J_2 &= \begin{bmatrix} 30 & 0 & 1 \\ 10 & 7,5 & 0 \\ 30 & 0 & 0 \end{bmatrix} \\
h_2 &= -J_2^{-1}F_2 = \begin{bmatrix} 0 \\ 0,0139 \\ 0 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
x_3 &= x_0 + h_0 = \begin{bmatrix} 0,1667 \\ 0,2639 \\ 95 \end{bmatrix} \\
&\vdots \\
F_5 &= 10^{-5} \begin{bmatrix} 0 \\ -0,2494 \\ 0 \end{bmatrix} \\
J_5 &= \begin{bmatrix} 30 & 0 & 1 \\ 10 & 7,0711 & 0 \\ 30 & 0 & 0 \end{bmatrix} \\
h_5 &= -J_5^{-1} F_5 = 10^{-6} \begin{bmatrix} 0 \\ 0,3527 \\ 0 \end{bmatrix} \\
x_6 &= x_0 + h_0 = \begin{bmatrix} 0,1667 \\ 0,2643 \\ 95 \end{bmatrix}
\end{aligned}
$$

onde x_6 é a solução do problema de intersecção, de acordo com a tolerância especificada.

5.3 Exercícios

Exercício 5.1 *Considere o exemplo 5.2, com $q_0 = (5,5,0)$, $q_1 = (15,0,0)$, $p_0 = (0,0,0)$, $p_1 = (0,0,-10)$ e $p_2 = (10,0,0)$. Explique o que acontece.*

Exercício 5.2 *Uma esfera de raio r e centro (c_x, c_y, c_z) pode ser definida, de forma paramétrica, como*

$$(r\cos(\theta)\cos(\phi) + c_x, r\cos(\theta)\text{sen}(\phi) + c_y, r\text{sen}(\theta) + c_z)$$

onde $-\pi/2 \leq \theta \leq \pi/2$ e $0 \leq \phi < 2\pi$. Calcule as intersecções dessa esfera com a reta $\overrightarrow{o} + t\overrightarrow{u}$, $\overrightarrow{o} = (10,10,10)$ e $\overrightarrow{u} = (-1,-1,-1)$. Utilize como estimativa inicial $\theta_0 = \phi_0 = t_0 = 0$ e uma tolerância de 10^{-5}.

Exercício 5.3 *Compare o processo numérico utilizado para resolver o exercício 5.2 com a solução do mesmo problema, obtida de forma algébrica. (*Dica: utilize a equação da reta na forma paramétrica $\overrightarrow{o} + t\overrightarrow{u}$ e substitua na equação implícita da esfera, $(x-c_x)^2 + (y-c_y)^2 + (z-c_z)^2 = r^2$, e isole t.*)*

Exercício 5.4 *Utilize a formulação apresentada no exemplo 5.3, para a reta $(100,20,5)+t(-1,0,0)$. Explique o que acontece.*

Exercício 5.5 *Resolva o sistema*

$$\begin{cases} 0,1x^2 - x + 0,1y^2 + 0,8 = 0 \\ 0,1x - y + 0,1xy^2 + 0,8 = 0 \end{cases}$$

sabendo que ele apresenta uma solução próxima a $(x,y) = (0,5,0,5)$.

Exercício 5.6 *Calcule (x,y) de modo que*

$$\begin{cases} x^2 + y^2 = 2 \\ x^2 - y^2 = 1 \end{cases}$$

usando $x_0 = y_0 = 1$.

Capítulo 6

Autovalores e Autovetores

6.1 Introdução

Neste capítulo, apresentaremos alguns dos métodos utilizados para a solução do *problema do autovalor*, i.e., o sistema de n equações lineares

$$Ax = \lambda x \tag{6.1}$$

para o qual procuramos um vetor solução x tal que $x_i \neq 0$ para pelo menos algum i, ou seja, uma solução não-trivial. Para que tal seja possível, é necessário que

$$\det(A - \lambda I) = 0 \tag{6.2}$$

a qual é uma equação polinomial de grau n na variável λ, chamada de *equação característica* de A; o polinômio $\det(A - \lambda I)$ é chamado de *polinômio característico* de A.

As n raízes de (6.2) são chamadas de *autovalores*, *raízes latentes* ou *valores característicos* de A. A cada raiz λ corresponde um vetor $x \in \mathbb{C}^n \neq 0$ que satisfaz a equação (6.1), o qual é chamado de *autovetor*, *vetor latente* ou *vetor característico* de A. Note que, se x é um autovetor de A, então kx, onde $k \in \mathbb{R}$, também é, pois

$$Akx = kAx = \lambda kx = k\lambda x.$$

Costumeiramente os autovetores são *normalizados*, i.e. $||x|| = 1$ em alguma norma escolhida (o que pode ser feito pela relação acima).

Se todas as raízes de (6.2) são distintas entre si, então isso implica em que a matriz A apresenta um *conjunto completo* de autovetores *linearmente independentes* (L.I.). No entanto, mesmo para casos em que os autovalores não são todos distintos, podemos encontrar um conjunto completo de autovetores L.I.

Podemos também calcular os autovalores da matriz inversa de A, A^{-1}, a partir dos autovalores de A. Se multiplicarmos a equação (6.1) à esquerda por A^{-1}, temos

$$x = \lambda A^{-1}x$$

ou

$$A^{-1}x = \frac{1}{\lambda}x. \tag{6.3}$$

Essa última equação nos diz que $\frac{1}{\lambda}$ é autovalor de A^{-1}, onde λ é um autovalor de A, com o autovetor x correspondente.

Problemas envolvendo autovalores e autovetores surgem em inúmeras aplicações, como podemos ver nos exemplos que seguem, conforme apresentados em [6].

Exemplo 6.1 *O estudo das vibrações de sistemas dinâmicos e de estruturas requer a solução de problemas de autovalores e autovetores. Considere, apenas para fins de explanação, o problema de se determinar as vibrações de pequenas partículas presas por um fio uniforme, sem peso, ao qual é aplicada uma força $\vec{F}$ nas extremidades (cf. a figura 6.1) e no qual desconsidera-se a ação da gravidade. As partículas encontram-se a distâncias iguais entre si e as vibrações das mesmas são consideradas pequenas e perpendiculares à posição de descanso do fio. Escrevendo as equações*

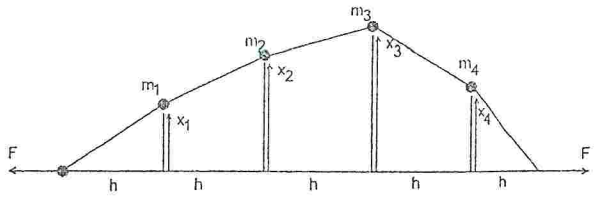

Figura 6.1: *O problema das vibrações.*

diferenciais para as forças atuantes em cada partícula, temos:

$$\begin{aligned}
m_1\frac{d^2x_1}{dt^2} &= -F\frac{x_1}{h} + F\frac{x_2 - x_1}{h} \\
m_2\frac{d^2x_2}{dt^2} &= -F\frac{x_2 - x_1}{h} + F\frac{x_3 - x_2}{h} \\
m_3\frac{d^2x_3}{dt^2} &= -F\frac{x_3 - x_2}{h} - F\frac{x_3 - x_4}{h} \\
m_4\frac{d^2x_4}{dt^2} &= +F\frac{x_3 - x_4}{h} - F\frac{x_4}{h}
\end{aligned}$$

Introduzindo a notação

$$\begin{aligned}
x &= (x_1, x_2, x_3, x_4)^T \\
d_i &= \frac{m_i h}{F}, \quad i = 1, 2, 3, 4
\end{aligned}$$

podemos escrever o sistema de equações diferenciais acima na forma matricial

$$D\frac{d^2x}{dt^2} = Tx \tag{6.4}$$

onde D é a matriz diagonal

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & d_3 & \\ & & & d_4 \end{bmatrix}$$

e T é a matriz tridiagonal

$$T = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}.$$

Quando as partículas vibram em fase ou em direções opostas, i.e., em modo normal, *então a condição*

$$\frac{d^2x}{dt^2} = -w^2x, \quad w \in \mathbb{R} \tag{6.5}$$

é satisfeita. Substituindo a equação (6.5) em (6.4), obtemos o problema de autovalor

$$Dw_i^2x_i = -Tx_i, \quad i = 1,2,3,4 \tag{6.6}$$

para as freqüências de vibração w_1, w_2, w_3 *e* w_4 *e os* modos normais *correspondentes, i.e., os autovetores* x_1, x_2, x_3 *e* x_4.

Aparentemente, se isolarmos x *no lado direito da equação (6.6), obteríamos o que se chama de* problema generalizado do autovalor, *cuja forma geral é*

$$(A - \lambda B)x = 0$$

onde A *e* B *são matrizes de ordem* n. *Porém, se introduzirmos o vetor*

$$y = D^{1/2}x$$

o que é possível, já que os elementos da diagonal de D *são positivos, por definição, então podemos escrever (6.6) como*

$$D^{-1/2}TD^{-1/2}y_i = -w_i^2y_i$$

o qual recai na forma 6.1.

Exemplo 6.2 *A teoria de Leontief sobre a compra e a venda de produtos é muito utilizada no campo de estudo da macroeconomia; como exemplo, consideramos as vendas e compras de produtos num setor industrial.*

Seja b_{ij} *as vendas da indústria* i *para a indústria* j; b_{ii} *representa os bens produzidos pela indústria* i *e retidos por ela própria. As vendas de bens da indústria* i *para o mercado é denotada por* y_i *e o total de bens produzidos por* x_i. *Então,*

$$x_i = y_i + \sum_j b_{ij} \tag{6.7}$$

A fim de definirmos b_{ij}, *assume-se que as vendas da indústria* i *para a* j *estão em proporção constante à produção da indústria* j, *i.e.*

$$b_{ij} = a_{ij}x_j$$

onde a_{ij} *são ditos* coeficientes de entrada. *Em uma situação estática, podemos escrever, a partir de (6.7),*

$$x = y + Ax \tag{6.8}$$

onde $x = (x_1, x_2, \ldots, x_n)^T$ *e* $y = (y_1, y_2, \ldots, y_n)^T$ *e* A *é matriz de ordem* n *cujos elementos* (i,j) *são os coeficientes de entrada* a_{ij}. *Ora, a equação (6.8) pode ser reescrita como*

$$(I - A)x = y \tag{6.9}$$

onde $I - A$ *é chamada de* matriz de Leontief. *A equação (6.9) pode ser resolvida calculando-se os autovalores e autovetores de* A. *Sua utilidade reside no fato de que, com ela, é possível determinar-se a quantidade de bens produzidos* (x) *necessários para satisfazer a uma demanda final* (y), *pré-estabelecida.*

Se a produção e a demanda não se encontram em equilíbrio, então devemos considerar um modelo dinâmico, que leve em consideração a taxa de variação da produção. Nesse caso, usualmente considera-se que a produção em cada indústria varia a uma taxa proporcional à diferença entre os níveis de venda e de produção. Daí,

$$\frac{dx(t)}{dt} = D\left((A - I)x(t) + y(t)\right) \tag{6.10}$$

onde D é uma matriz diagonal de ordem n, cujos elementos d_{ii} representam os coeficientes de reação *das indústrias.*

A equação (6.10) substitui nesse caso a equação (6.8) e representa o comportamento dinâmico do sistema econômico em estudo. Uma das questões a serem estudadas, nesse caso, é se o sistema é estável, determinando-se os autovalores e autovetores da matriz $D(A-I)$. Particularmente, procura-se determinar se esses autovalores tem parte real positiva pois, como as soluções do sistema de equações diferenciais (6.10) são da forma $e^{\lambda_i t}$, isso indicaria uma instabilidade, já que a demanda $x(t)$ cresceria exponencialmente com o tempo.

A seguir, apresentaremos dois importantes teoremas, os quais nos permitirão desenvolver técnicas de determinação de autovalores e autovetores para um tipo específico de matrizes.

6.2 Teoremas de limites sobre autovalores

Teorema 6.2.1 *Discos de Gerschgorin: Seja A uma matriz de ordem n, e d_i, $i = 1, 2, \ldots, n$ os discos cujos centros são os elementos a_{ii} e cujos raios r_i são dados por*

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad i = 1, 2, \ldots, n.$$

Seja D a união de todos os discos d_i. Então, todos os autovalores de A encontram-se contidos em D.

Prova: Seja λ um autovalor de A e x um autovetor correspondente, tal que $\max_i |x_i| = 1$. Então,

$$\lambda x = Ax$$

de onde

$$(\lambda - a_{ii})x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j, \quad i = 1, 2, \ldots n$$

Supondo que $|x_k| = 1$, então

$$\begin{aligned} |\lambda - a_{kk}| &\leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{kj}||x_j| \\ &\leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{kj}| = r_k \end{aligned}$$

i.e., o autovalor λ está contido no disco d_k e, como λ é arbitrário, então todos os autovalores de A devem estar contidos na união de todos os discos, D. $\Diamond$

O exemplo a seguir apresenta uma aplicação do teorema 6.2.1.

Exemplo 6.3 *A matriz*

$$A = \begin{bmatrix} -1 & 0 & -1 \\ -1 & 4 & -1 \\ -1 & -2 & 10 \end{bmatrix}$$

tem como seus autovalores $\lambda_1 = 10,3863$, $\lambda_2 = 3,8037$ e $\lambda_3 = 0,8100$. Calculando os discos de Gerschgorin, temos:

$$\begin{aligned} d_1 &= |z-1| < |0| + |-1| = 1 \\ d_2 &= |z-4| < |-1| + |-1| = 2 \\ d_3 &= |z-10| < |-1| + |-2| = 3 \end{aligned}$$

Como todos os autovalores de A são reais, e observando (veja a figura 6.2) que em cada disco devemos ter um autovalor, podemos dizer que:

- *existe um autovalor, λ_1, que está dentro do disco centrado em 10 e raio 3 e, de fato,* $7 < 10,3863 < 13$;
- *existe um autovalor, λ_2, que está dentro do disco centrado em 4 e raio 2 e, realmente,* $2 < 3,8037 < 6$;
- *existe um autovalor, λ_3, que está dentro do disco centrado em 1 e raio 1 e, com efeito,* $0 < 0,81 < 2$;

A figura 6.2 ilustra esse resultado.

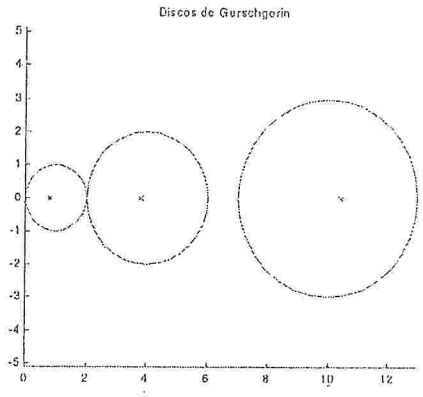

Figura 6.2: *Discos de Gerschgorin*

Uma conseqüência do teorema de Gerschgorin é a determinação do maior disco que contém todos os autovalores de A. Podemos obter, a partir dos discos, os extremos ao longo do eixo dos números reais, i.e. o intervalo $[\alpha, \omega]$ tal que

$$\alpha = \min_i \{a_{ii} - r_i\}, \quad \omega = \max_i \{a_{ii} + r_i\}, \quad i = 1, 2, \ldots, n \tag{6.11}$$

e o maior disco é justamente aquele com centro $(\alpha + \omega)/2$ e raio $(\alpha + \omega)/2$. No caso em que todos os autovalores são reais, basta então considerar o intervalo $[\alpha, \omega]$.

Teorema 6.2.2 *Maior e menor autovalor: Seja A uma matriz real simétrica de ordem n, e* $x \in \mathbb{C}$ *um vetor arbitrário. Então,*

$$\lambda_1 = \max_{x \neq 0} \frac{x^T A x}{x^T x}, \quad \lambda_n = \min_{x \neq 0} \frac{x^T A x}{x^T x}$$

onde os autovalores são ordenados tais que $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$.

A razão

$$\frac{x^T A x}{x^T x}, \quad x \neq 0 \tag{6.12}$$

é chamada de *quociente de Rayleigh* correspondente a x e, juntamente com o teorema 6.2.2, nos permitirá estimar de forma bastante rápida um autovalor de uma matriz simétrica, conforme veremos na seção 6.5.

6.3 Cálculo de autovalores e autovetores via determinantes

Já vimos que, por definição, os autovalores de uma matriz A são as raízes do polinômio característico de A. Evidentemente, para matrizes de ordem $n > 4$, não é aconselhável que se utilize a equação (6.2) para se obter o polinômio característico, por duas razões:

1. o cálculo de determinantes de ordem superior a 4 envolve considerável custo computacional;
2. o polinômio característico de uma matriz grande pode ser instável numericamente.

No entanto, algumas aplicações de engenharia, física e outros campos do conhecimento envolvem a determinação de autovalores de matrizes de ordem $n = 2$ ou $n = 3$ e, nesse caso, é possível obter-se os autovalores extraindo as raízes do polinômio característico, conforme mostra o exemplo a seguir.

Exemplo 6.4 *Seja a matriz*

$$\begin{bmatrix} 2 & 5 \\ 3 & -4 \end{bmatrix}.$$

O seu polinômio característico é

$$p(\lambda) = \det(A - \lambda I) = \begin{vmatrix} 2-\lambda & 5 \\ 3 & -4-\lambda \end{vmatrix} = (2-\lambda)(-4-\lambda) - 15$$

ou $p(\lambda) = \lambda^2 + 2\lambda - 23$, *cujas raízes são* $\lambda_1 = 3,8990$ *e* $\lambda_2 = -5,8990$.

Para se determinar os autovetores, utiliza-se a equação (6.1) para cada autovalor λ_i, na forma $(A - \lambda_i I)x_i = 0$, como segue:

Exemplo 6.5 *Calcule os autovetores do exemplo 6.4.*
Solução: *Para o autovalor* $\lambda_1 = 3,8990$, *escrevemos*

$$\begin{aligned} (A - 3,8990I)x_1 &= 0 \\ \begin{bmatrix} 7,8990 & 5 \\ 3 & 1,8990 \end{bmatrix} \begin{bmatrix} (x_1)_1 \\ (x_1)_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned}$$

de onde obtemos

$$x_1 = \begin{bmatrix} k \\ -1,5798k \end{bmatrix}, \quad k \neq 0$$

O autovetor correspondente a $\lambda_2 = -5,8990$ *é obtido de forma similar:*

$$\begin{aligned} (A + 5,8990I)x_1 &= 0 \\ \begin{bmatrix} -1,8990 & 5 \\ 3 & -7,8990 \end{bmatrix} \begin{bmatrix} (x_2)_1 \\ (x_2)_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned}$$

de onde obtemos

$$x_2 = \begin{bmatrix} k \\ 0,3798k \end{bmatrix}, \quad k \neq 0$$

Computacionalmente, no entanto, podemos estimar o autovetor correspondente a um autovalor utilizando os métodos da *potência com translação da origem* (seção 6.5.2) ou da *iteração inversa com translação da origem* (seção 6.5.3).

6.4 Autovalores de uma matriz tridiagonal simétrica

Em muitas aplicações surgem matrizes *tridiagonais simétricas*, das quais necessitamos extrair autovalores e/ou autovetores. Por exemplo, ao aproximarmos a equação diferencial parcial

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

por diferenças finitas, obtemos uma matriz

$$\begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}$$

a qual apresenta aquela característica.

De forma geral, consideramos uma matriz T de ordem n,

$$T = \begin{bmatrix} a_1 & b_1 & 0 & \dots & 0 \\ b_1 & a_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_{n-1} \\ 0 & \dots & 0 & b_{n-1} & a_n \end{bmatrix} \tag{6.13}$$

e chamamos de T_r a matriz principal de ordem r de T, i.e.

$$T_1 = \begin{bmatrix} a_1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} a_1 & b_1 \\ b_1 & a_2 \end{bmatrix}, \quad T_3 = \begin{bmatrix} a_1 & b_1 & 0 \\ b_1 & a_2 & b_2 \\ 0 & b_2 & a_3 \end{bmatrix}, \quad \dots$$

Escrevendo as equações características $p_1(\lambda)$, $p_2(\lambda)$ e $p_3(\lambda)$ das matrizes T_1, T_2 e T_3, obtemos:

$$p_1(\lambda) = \det(T_1 - \lambda I) = a_1 - \lambda \tag{6.14}$$

$$p_2(\lambda) = \det(T_2 - \lambda I) = (a_2 - \lambda)(a_1 - \lambda) - b_1^2 = (a_2 - \lambda)p_1(\lambda) - b_1^2 \tag{6.15}$$

$$\begin{aligned} p_3(\lambda) &= \det(T_3 - \lambda I) = (a_3 - \lambda)\left((a_2 - \lambda)(a_1 - \lambda) - b_1^2(a_3 - \lambda)\right) - b_2^2(a_1 - \lambda) = \\ &= (a_3 - \lambda)p_2(\lambda) - b_2^2 p_1(\lambda) \end{aligned} \tag{6.16}$$

de onde podemos escrever, generalizando para r,

$$p_r(\lambda) = (a_r - \lambda)p_{r-1}(\lambda) - b_{r-1}^2 p_{r-2}(\lambda), \quad r = 2, 3, \dots, n, \quad p_0(\lambda) = 1, \tag{6.17}$$

A equação (6.17) nos permite avaliar o polinômio característico da matriz T de forma bastante eficiente; no entanto, estamos preocupados em obter os autovalores de T, ou as raízes de p_n. O teorema a seguir nos permitirá escrever um algoritmo bastante eficiente para se extrair alguns ou todos os autovalores de T.

Teorema 6.4.1 *Seqüência de Sturm: Se a matriz tridiagonal (6.13) é não-reduzível* [1]*, então os $r-1$ autovalores μ da matriz T_{r-1} separam estritamente os r autovalores λ da matriz T_r:*

$$\lambda_r < \mu_{r-1} < \lambda_{r-1} < \mu_{r-2} < \cdots < \lambda_2 < \mu_1 < \lambda_1.$$

Mais ainda, se $s(\lambda)$ representa o número de trocas de sinal na seqüência

$$\{p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)\}$$

então $s(\lambda)$ é igual ao número de autovalores de T menores do que λ, onde $p_r(\lambda)$ é dado por (6.17) e assume-se que $p_r(\lambda)$ tem o sinal oposto de $p_{r-1}(\lambda)$ se $p_r(\lambda) = 0$.

[1] Uma matriz A é dita não-reduzível se os elementos da diagonal da matriz triangular superior R, resultante de sua fatoração no produto QR, são todos não-nulos, onde Q é uma matriz ortogonal (i.e. $Q^T Q = I$).

O teorema 6.4.1 é extremamente importante: ele nos diz que, se tivermos os n autovalores μ de uma matriz triadiagonal T_n (de ordem n), então entre cada par de autovalores consecutivos μ (com exceção do menor e do maior), existe *um e apenas um* autovalor λ da matriz tridiagonal T_{n+1} (de ordem $n+1$), obtida acrescentando-se uma linha e uma coluna à matriz T_n. Devido à essa característica, podemos utilizar o algoritmo da bissecção (ver algoritmo 2.2.1), juntamente com a equação (6.17), para obtermos rapidamente, e com segurança, um autovalor de T_{n+1}, à partir de um intervalo que é um par de autovalores consecutivos de T_n.

Para obter-se o menor e o maior autovalores de T_{n+1}, utilizamos o teorema de Gerschgorin - mais especificamente, calculamos o maior intervalo que engloba todos os autovalores, com a equação (6.11). Assim, o menor autovalor é calculado usando-se como estimativa inicial para o método da bissecção o intervalo $[\alpha, \mu_{r-1}]$; para o maior autovalor, utiliza-se o intervalo $[\mu_1, \omega]$.

Os algoritmos 6.4.1, 6.4.2 e 6.4.3 combinam as idéias apresentadas acima. Da maneira como o algoritmo 6.4.3 é apresentado, todos os autovalores são obtidos; no entanto, simples modificações do mesmo nos permitem obter apenas alguns autovalores (por exemplo, o maior e o menor, ou os dois maiores, etc.).

O exemplo 6.6 demonstra uma situação típica, resolvido utilizando-se esses algoritmos.

Algoritmo 6.4.1 *Avalia polinômio característico de uma matriz tridiagonal simétrica*

```
function pol_carac_trid(input: x, a, b; output: p)
        % a e b são os vetores contendo os elementos da
        % diagonal e subdiagonal, respectivamente,
        % da matriz tridiagonal
        p_0 ← 1
        p_1 ← a_1 − x
        p ← p_1
        for r ← 2, 3, ..., n do
            p ← (a_r − x)p_1 − b²_{r−1} p_0
            p_0 ← p_1
            p_1 ← p
        endfor
endfunction
```

Algoritmo 6.4.2 *Método da bissecção com polinômio característico*

```
proc bissecção_trid(input: a, b, α, β, k_max, δ, ϵ; output: χ)
    % a e b são os vetores contendo os elementos da
    % diagonal e subdiagonal, respectivamente,
    % da matriz tridiagonal
    u ← pol_carac_trid(α, a, b)
    v ← pol_carac_trid(β, a, b)
    e ← β − α
    if (sign(u) = sign(v)) then
      "não pode proceder"
    else
      k ← 1
      w ← 1
      while ((k ≤ k_max) AND (|e| ≥ δ) AND (|w| ≥ ϵ))
          e ← e/2
          χ ← α + e
          w ← pol_carac_trid(χ, a, b)
          if (sign(w) ≠ sign(u)) then
            β ← χ
            v ← w
          else
            α ← χ
            u ← w
          endif
          k ← k + 1
      endwhile
    endif
endproc
```

Algoritmo 6.4.3 *Autovalores de uma matriz tridiagonal simétrica*

proc autovalores_tridiagonal(input: a, b, n; *output:* λ)
  % *Calcula os raios dos discos de Gerschgorin, cada qual com centro* $a(i)$
  $r_1 \leftarrow |b_1|$
  for $i \leftarrow 2, 3, \ldots, n-1$ *do*
    $r_i \leftarrow |b_{i-1}| + |b_i|$
  endfor
  $r_n \leftarrow |b_{n-1}|$
  % *Calcula o intervalo* $[\alpha, \omega]$ *na reta dos reais*
  % *contendo os autovalores*
  $\alpha \leftarrow \min_{i=1}^{n}(a_i - r_i)$
  $\omega \leftarrow \max_{i=1}^{n}(a_i + r_i)$
  % *Calcula os autovalores, iniciando com o autovalor*
  % *de* $T_1 = [a_1]$, $\mu = a_1$
  $\mu_1 = a_1$
  for $i \leftarrow 2, 3, \ldots, n$ *do*
    % *Calcula os autovalores de* T_i
    % *a. entre* α *e* μ_1
    call bissecção_trid$(a, b, \alpha, \mu_1, k_{max}, \delta, \epsilon, \lambda_1)$
    % *b. autovalores entre* μ_1 *e* μ_{i-1}
    for $j \leftarrow 1, 2, \ldots, i-2$ *do*
      call bissecção_trid$(a, b, \mu_j, \mu_{j+1}, k_{max}, \delta, \epsilon, \lambda_{j+1})$
    endfor
    % *c. entre* μ_{i-1} *e* ω
    call bissecção_trid$(a, b, \mu_{i-1}, \omega, k_{max}, \delta, \epsilon, \lambda_1)$
    $\mu \leftarrow \lambda$
  endfor
  $\lambda \leftarrow \mu$
endproc

Exemplo 6.6 *Seja a matriz tridiagonal*

$$T = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix}$$

a qual pode ser representada de forma compacta através dos vetores

$$a = (2, 2, 2, 2)^T \quad e \quad b = (-1, -1, -1)^T$$

Para se obter os autovalores de T, *iniciamos com a matriz*

$$T_1 = [\ a_1\ ] = [\ 2\ ]$$

a qual tem como seu único autovalor $\mu_1 = a_1 = 2$. *Além disso, calculamos os extremos do intervalo de Gerschgorin,* $\alpha = 0$ *e* $\omega = 6$, *através da equação (6.11).*

Agora, precisamos calcular os dois autovalores λ *da matriz*

$$T_2 = \begin{bmatrix} a_1 & b_1 \\ b_1 & a_2 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

os quais, pelos teoremas de Gerschgorin e 6.4.1, satisfazem

$$\alpha < \lambda_2 < \mu_1 < \lambda_1 < \omega$$

Estipulando-se como tolerâncias de convergência para o método da bisseção $\delta = \epsilon = \sqrt{\varepsilon}$ *e um máximo de* 200 *iterações,* λ_2 *é obtido em* 2 *iterações utilizando-se como intervalo de busca* $[\alpha, \mu_1] = [0, 2]$, *resultando no valor* $\lambda_2 = 1$. *O autovalor* λ_1 *também é obtido em* 2 *iterações, usando-se como intervalo de busca* $[\mu_1, \beta] = [2, 6]$, *com o qual obtém-se* $\lambda_1 = 3$.

Antes de procedermos ao cálculo dos autovalores de T_3, *fazemos uma cópia dos* λ, *armazenando-os em* μ; *assim, temos* $\mu_1 = 3$ *e* $\mu_2 = 1$.

Procedemos, então, com o cálculo dos autovalores λ *de* T_3; *para tanto, utilizamos os intervalos de busca*

- $[\alpha, \mu_2] = [0, 1]$ *para calcular o autovalor* λ_3;
- $[\mu_2, \mu_1] = [1, 3]$ *para calcular o autovalor* λ_2;
- $[\mu_1, \omega] = [3, 6]$ *para calcular o autovalor* λ_1.

Os autovalores λ_3, λ_2 *e* λ_1 *são então obtidos com o método da bissecção, utilizando-se as mesmas tolerâncias especificadas anteriormente, resultando em* $\lambda_3 = 0,5858$, $\lambda_2 = 2$ *e* $\lambda_1 = 3,4142$, *obtidos em* 27, 2 *e* 27 *iterações respectivamente.*

Finalmente, basta calcularmos os autovalores de T_4. *Procedendo de forma similar, fazemos* $\mu_3 = 0,5858$, $\mu_2 = 2$ *e* $\mu_1 = 3,4142$ *e estipulamos os intervalos de busca*

- $[\alpha, \mu_3] = [0, 0, 5858]$ *para calcular o autovalor* λ_4;
- $[\mu_3, \mu_2] = [0,5858, 2]$ *para calcular o autovalor* λ_3;
- $[\mu_2, \mu_1] = [2, 3, 4142]$ *para calcular o autovalor* λ_2;
- $[\mu_1, \omega] = [3, 4142, 6]$ *para calcular o autovalor* λ_1.

de onde, após aplicarmos o algoritmo da bissecção a cada um desses intervalos, obtemos os autovalores de $T_4 \equiv T$, $\lambda_4 = 0,3820$, $\lambda_3 = 1,3820$, $\lambda_2 = 2,6180$ *e* $\lambda_1 = 3,6180$, *após* 27 *iterações (para todos os intervalos de busca).*

Note que não se obtém, com essa técnica, os autovetores correspondentes aos autovalores. Os métodos apresentados na seção a seguir podem ser utilizados para se obter esses autovetores.

6.5 Métodos para aproximação de autovalores e autovetores

Em muitas aplicações, não é necessário obter-se todos os autovalores; é comum desejar-se, por exemplo, obter apenas o maior autovalor e seu correspondente autovetor. Os métodos apresentados nessa seção são indicados para o caso em que apenas um dos autovalores (e seu autovetor) necessita ser calculado. Particularmente, tais métodos são *iterativos* e apresentam boa eficiência quando a matriz em estudo é grande, esparsa e apresenta uma grande separação relativa entre o autovalor desejado e os demais autovalores.

6.5.1 Método da potência

Seja A uma matriz de ordem n com autovalores λ_i tais que

$$|\lambda_1| = |\lambda_2| = \ldots = |\lambda_r| > |\lambda_{r+1}| \geq \ldots \geq |\lambda_n| \tag{6.18}$$

Nesse caso, diz-se que A apresenta r autovalores *dominantes*. Por hipótese, assumimos que existem n autovetores linearmente independentes x_i, de onde qualquer vetor arbitrário z_0 pode ser expresso como combinação linear desses autovetores, i.e.

$$z_0 = \sum_{i=1}^{n} \alpha_i x_i \tag{6.19}$$

Considere agora o método de aproximação sucessiva

$$z_k = Az_{k-1}, \quad k = 1, 2, \ldots \tag{6.20}$$

onde z_0 é um valor inicial, dado. Usando as equações (6.1), (6.19) e escrevendo (6.20) em termos de z_0, temos

$$\begin{aligned} z_k &= Az_{k-1} = A^2 z_{k-2} = \ldots = A^k z_0 \\ &= \sum_{i=1}^{n} \alpha_i \lambda_i^k x_i \end{aligned} \tag{6.21}$$

Se pelo menos um dos α_1, α_2, ..., α_r não é nulo, então os termos correspondentes a eles, i.e. $\sum_{i=1}^{r} \alpha_i \lambda_i^k x_i$ irão dominar o somatório da equação (6.21).

Suponha, por exemplo, que temos um autovalor dominante, λ_1, de A. Considerando que $\alpha_1 \neq 0$, podemos reescrever (6.21) como

$$z_k = \lambda_1^k \left(\alpha_1 x_1 + \sum_{i=1}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i \right)$$

Note agora que, como $\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n$, por hipótese, então os termos

$$\left(\frac{\lambda_i}{\lambda_1} \right)^k$$

tendem a zero à medida que k cresce. Daí, podemos escrever

$$z_k = \lambda_1^k (\alpha_1 x_1 + \epsilon_k) \tag{6.22}$$

onde ϵ_k é um vetor com elementos próximos a zero. O vetor z_k tende, então, a aproximar o autovetor não-normalizado x_1. Essa equação nos permite escrever o assim chamado *método da potência*.

Da equação (6.22), podemos escrever

$$z_{k+1} = \lambda_1^{k+1} (\alpha_1 x_1 + \epsilon_{k+1})$$

e, dividindo a i-ésima componente da equação acima pela componente correspondente de (6.22), obtemos

$$\frac{(z_{k+1})_i}{(z_k)_i} = \lambda_1 \left(\frac{\alpha_1 x_1 + \epsilon_{k+1}}{\alpha_1 x_1 + \epsilon_k} \right) \rightarrow \lambda_1, \quad \text{quando } k \rightarrow \infty, \quad i = 1, 2, \ldots, n \tag{6.23}$$

onde $(z_k)_i$ indica o elemento i do vetor z_k. A equação (6.23) nos diz que a taxa de convergência do método depende não só das constantes α_i, mas principalmente das frações

$$\left| \frac{\lambda_2}{\lambda_1} \right|, \left| \frac{\lambda_3}{\lambda_1} \right|, \ldots, \left| \frac{\lambda_n}{\lambda_1} \right|.$$

Quanto menores forem esses frações, mais rápida é a convergência; por isso diz-se que o método da potência é eficiente – converge rapidamente para um autovalor – *desde que este autovalor seja dominante*, i.e., relativamente distante dos demais.

De posse das equações (6.20) e (6.23), podemos escrever um algoritmo para o método da potência. Uma questão que se coloca é: quais valores iniciais, λ_0 e z_0, devemos utilizar para o autovalor dominante e seu autovetor? Para z_0, consideraremos um vetor arbitrário, o qual será normalizado antes de se iniciar as iterações. Com essa escolha, valemo-nos da equação (6.1) e escrevemos

$$\begin{aligned} Az_0 &= \lambda_0 z_0 \therefore \| z_0 \| = 1 \therefore \\ z_0^T Az_0 &= z_0^T \lambda_0 z_0 = \lambda_0 (z_0^T z_0) = \lambda_0 \end{aligned}$$

Aplica-se, então, repetidamente a equação (6.20), normalizando o vetor z_k a cada iteração, conforme mostrado no algoritmo 6.5.1. O exemplo 6.7 mostra o funcionamento do método.

Algoritmo 6.5.1 *Método da potência*

proc potencia(input: A, z_0, ϵ, $k_{\max}$; *output:* λ_k, z_k)
  $z_0 \leftarrow z_0/\|z_0\|$
  $\lambda_0 \leftarrow z_0^T A z_0$
  for $k = 1, 2, \ldots, k_{\max}$ *do*
    $q \leftarrow A z_{k-1}$
    $z_k \leftarrow q/\|q\|$
    $\lambda_k \leftarrow z_k^T A z_k$
    if $|\lambda_k - \lambda_{k-1}| < \epsilon$ *then*
      break
    endif
  endfor
endproc

Exemplo 6.7 *Seja a matriz*

$$A = \begin{bmatrix} 8 & 1 & 2 \\ -1 & 5 & 1 \\ 0 & 1 & 90 \end{bmatrix},$$

a qual tem como autovalores e respectivos autovetores,

$$\lambda_1 = 90,0115, \quad \lambda_2 = 7,6308, \quad \lambda_3 = 5,3577$$

$$x_1 = \begin{bmatrix} 0,0245 \\ 0,0115 \\ 0,9996 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -0,9353 \\ 0,3539 \\ -0,0043 \end{bmatrix}, \quad x_3 = \begin{bmatrix} -0,0043 \\ -0,0111 \\ 0,9996 \end{bmatrix}.$$

Utilizando-se o método da potência com um vetor com três elementos escolhidos arbitrariamente e normalizado, $z_0 = (0,4394, 0,6415, 0,6287)^T$, *obtém-se a seguinte seqüência de valores, com uma tolerância para convergência de* 10^{-5}*:*

k	z_k	λ_k
0	$(0,4394, 0,6415, 0,6287)^T$	40,5408
1	$(0,0940, 0,0590, 0,9938)^T$	89,2834
2	$(0,0313, 0,0133, 0,9994)^T$	89,9939
3	$(0,0251, 0,0115, 0,9996)^T$	90,0102
4	$(0,0246, 0,0115, 0,9996)^T$	90,0114
5	$(0,0245, 0,0115, 0,9996)^T$	90,0115
6	$(0,0245, 0,0115, 0,9996)^T$	90,0115

onde pode-se verificar que z_6 *é uma boa aproximação para* x_1, *sujeita àquela tolerância.*

Caso a matriz tenha autovalores dominantes repetidos, i.e.

$$\lambda_1 = \lambda_2 = \ldots = \lambda_r$$

o método da potência irá obter apenas *um* autovetor, o qual será combinação linear dos autovetores correspondentes a λ_1.

O método da potência diverge se A tiver autovalores diferentes, porém de mesmo valor absoluto como, por exemplo, um par conjugado de autovalores dominantes complexos, $\lambda_2 = \overline{\lambda_1}$; o exemplo a seguir ilustra tal situação:

Exemplo 6.8 *Seja a matriz*

$$A = \begin{bmatrix} 1 & 10 & 2 \\ -1 & 1 & 10 \\ 10 & 1 & -13 \end{bmatrix},$$

a qual tem como autovalores e respectivos autovetores,

$$\lambda_1 = -9,4515 + 3,8807i, \quad \lambda_2 = -9,4515 - 3,8807i, \quad \lambda_3 = 7,9030$$

$$x_1 = \begin{bmatrix} 0,0307 - 0,4143i \\ 0,2160 + 0,5518i \\ -0,4368 - 0,5343i \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0,0307 + 0,4143i \\ 0,2160 - 0,5518i \\ -0,4368 + 0,5343i \end{bmatrix}, \quad x_3 = \begin{bmatrix} 0,7897 \\ 0,4651 \\ 0,4000 \end{bmatrix}.$$

Utilizando-se o método da potência com um vetor com três elementos escolhidos arbitrariamente e normalizado, $z_0 = (0,4857, 0,0197, 0,8739)^T$, *obtém-se a seguinte seqüência de valores, com uma tolerância para convergência de* 10^{-5}*:*

k	z_k	λ_k
0	$(0,4857, 0,0197, 0,8739)^T$	$-4,3241$
1	$(0,2253, 0,7668, -0,6010)^T$	$-9,1975$
2	$(0,4829, -0,3946, 0,7817)^T$	$-8,1345$
3	$(-0,2066, 0,7546, -0,6229)^T$	$-9,4601$
4	$(0,5786, -0,5001, 0,6443)^T$	$-6,4876$
5	$(-0,4517, 0,7731, -0,4454)^T$	$-6,2931$
6	$(0,8581, -0,4338, 0,2749)^T$	$-1,8886$

a qual apresenta um comportamento não convergente.

6.5.2 O método da potência com translação da origem

Como vimos na seção anterior, a convergência do método da potência depende de $|\lambda_2/\lambda_1|$, para o caso de existir apenas um autovalor dominante. Se essa razão for muito próxima de 1, então a convergência é muito lenta.

No entanto, podemos obter a solução de forma mais rápida se procedermos a uma modificação do método da potência. Essa modificação baseia-se no fato de que, se λ é um autovalor de uma matriz A, então $\lambda - \sigma$ é o autovalor correspondente da matriz $A - \sigma I$. Dessa forma, se aplicarmos o método da potência a uma matriz $A - \sigma I$, tal que $\lambda_1 - \sigma$ ainda seja dominante, a convergência do método dependerá de

$$\left| \frac{\lambda_2 - \sigma}{\lambda_1 - \sigma} \right|$$

o que, para um valor adequado de σ, poderá ser menor do que $|\lambda_2/\lambda_1|$. A esse processo, dá-se o nome de *translação da origem* – é como se os autovalores estivessem distribuídos em um novo sistema de referência cuja origem é σ, e não mais zero – e pode ser bastante eficaz, desde que a escolha de σ seja criteriosa.

Obviamente, poderíamos calcular explicitamente a matriz $A - \sigma I$ e utilizar o algoritmo 6.5.1; no entanto, pequenas modificações naquele algoritmo nos permitem utilizar o processo de translação da origem de forma mais eficiente. Novamente, z_0 é considerado um vetor unitário, de onde podemos obter a seguinte estimativa para λ_0:

$$\begin{aligned} (A - \sigma I)z_0 &= \lambda_0 z_0; \quad \text{pré-multiplicando por } z_0^T, \\ z_0^T A z_0 - \sigma z_0^T z_0 &= \lambda_0 (z_0^T z_0) \therefore \| z_0 \| = 1 \therefore \\ \lambda_0 &= z_0^T A z_0 - \sigma \end{aligned}$$

e, por analogia, escrevemos

$$\lambda_k = z_k^T A z_k - \sigma$$

Além disso, ao invés de calcularmos $q = Az_k$, devemos calcular $q = Az_k - \sigma z_k$. Ao final do processo, devemos corrigir λ_k, adicionando a ele σ. Essas idéias são apresentadas no algoritmo 6.5.2.

Algoritmo 6.5.2 *Método da potência (com translação)*

proc potencia_translação(input: A, z_0, σ, ϵ, $k_{\max}$; *output:* λ_k, z_k)
  $z_0 \leftarrow z_0/\| z_0 \|$
  $\lambda_0 \leftarrow z_0^T A z_0 - \sigma$
  for $k = 1, 2, \ldots, k_{\max}$ *do*
    $q \leftarrow A z_{k-1} - \sigma z_{k-1}$
    $z_k \leftarrow q/\| q \|$
    $\lambda_k \leftarrow z_k^T A z_k - \sigma$
    if $| \lambda_k - \lambda_{k-1} | < \epsilon$ *then*
      break
    endif
  endfor
  $\lambda_k \leftarrow \lambda_k + \sigma$
endproc

O exemplo a seguir ilustra o uso do método da potência com translação de origem.

Exemplo 6.9 *Seja a matriz*

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix},$$

a qual tem como autovalores e respectivos autovetores,

$$\lambda_1 = 3,4142, \quad \lambda_2 = 2, \quad \lambda_3 = 0,5858$$

$$x_1 = \begin{bmatrix} 0,5000 \\ -0,7071 \\ 0,5000 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0,7071 \\ 0,0000 \\ -0,7071 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 0,5000 \\ 0,7071 \\ 0,5000 \end{bmatrix}.$$

Note que $| \lambda_2/\lambda_1 | = 0,5858$. *Se utilizarmos o método da potência, a uma tolerância de* 10^{-5} *e vetor inicial* $z_0 = (1,0,0)^T$, *necessitaremos de* 13 *iterações para obter a aproximação* $3,4142$ *para o autovalor* λ_1.

No entanto, se usarmos o translação da origem, com $\sigma = 1$, *necessitamos apenas de* 9 *iterações para obter a mesma aproximação; veja que*

$$\left| \frac{\lambda_2 - 1}{\lambda_1 - 1} \right| = \left| \frac{1}{2,4142} \right| = 0,4142 < 0,5858 = \left| \frac{\lambda_2}{\lambda_1} \right|$$

o que sugere o menor número de iterações.

6.5.3 Método da iteração inversa

Como vimos, o método da potência nos permite aproximar o autovalor dominante de A; suponha, agora, que desejamos aproximar o *menor* autovalor (e seu correspondente autovetor) de A. Relembrando que os autovalores de A^{-1} são o inverso dos autovalores de A (equação (6.3)), então, se utilizarmos o método da potência sobre a matriz A^{-1}, aproximaremos o *menor* autovalor de A pois ele é o *maior* autovalor de A^{-1}. A essa modificação do método da potência chamamos de *método da iteração inversa.*

O método da iteração inversa procede, basicamente, com o cálculo sucessivo de vetores z_k dados por

$$z_k = A^{-1} z_{k-1}, \quad k = 1, 2, \ldots$$

mas já vimos (capítulo 4) que, computacionalmente, devemos evitar, se possível, calcular a inversa de uma matriz. Nesse caso, é aconselhado que se resolva o sistema

$$A z_k = z_{k-1}, \quad k = 1, 2, \ldots$$

através da fatoração LU de A (seção 4.3.2), uma vez que várias iterações serão necessárias para se aproximar o menor autovalor e respectivo autovetor.

Além disso, o método da iteração inversa é, normalmente, combinado com a translação de origem, o que resulta na fatoração LU da matriz $A - \sigma I$. O algoritmo 6.5.3 apresenta o método da iteração inversa, incorporando translação de origem.

Algoritmo 6.5.3 *Método da iteração inversa (com translação)*

proc iteração_inversa_translação(input: A, z_0, σ, ϵ, $k_{\max}$;
output: λ_k, z_k)
Fatore $A - \sigma I$ no produto LU
$z_0 \leftarrow z_0 / \| z_0 \|$
$\lambda_0 \leftarrow z_0^T A z_0 - \sigma$
for $k = 0, 1, \ldots, k_{\max}$ *do*
Resolva o sistema $Ly = z_k$
Resolva o sistema $Uq = y$
$z_k \leftarrow q / \| q \|$
$\lambda_k \leftarrow z_k^T A z_k - \sigma$
if $| \lambda_k - \lambda_{k-1} | < \epsilon$ *then*
break
endif
endfor
$\lambda_k \leftarrow \lambda_k + \sigma$
endproc

Exemplo 6.10 *Seja a matriz do exemplo 6.9,*

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix},$$

cujo menor autovalor é $\lambda_3 = 0,5858$ e o seu correspondente autvetor é $x_3 = (0,5000, 0,7071, 0,5000)^T$.

Se utilizarmos o algoritmo 6.5.3 com $\sigma = 0$, i.e. sem translação da origem, a uma tolerância de 10^{-5} e vetor inicial $z_0 = (1, 0, 0)^T$, necessitaremos de 7 iterações para obter a aproximação $0,5858$ para o autovalor λ_3 e $(0,5002, 0,7071, 0,4998)^T$ para o correspondente autovetor.

No entanto, se usarmos o translação da origem, com $\sigma = 0,5$, necessitamos apenas de 4 iterações para obter a mesma aproximação; veja que

$$\left| \frac{\lambda_2^{-1} - 0,5}{\lambda_3^{-1} - 0,5} \right| = \left| \frac{0}{0,0858} \right| = 0 < 0,2929 = \left| \frac{\lambda_2^{-1}}{\lambda_3^{-1}} \right|$$

o que sugere o menor número de iterações; na verdade, $\sigma = 0,5$ é a melhor escolha possível, nesse caso.

Outro exemplo mostra como usar o método da iteração inversa em conjunto com o teorema de Gerschgorin, a fim de se determinar um autovalor específico.

Exemplo 6.11 *Seja a matriz*

$$A = \begin{bmatrix} 5 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Os discos de Gerschgorin são:

$$\begin{aligned} d_1 &: c_1 = 5, \quad r_1 = 3 \\ d_2 &: c_2 = 3, \quad r_2 = 3 \\ d_3 &: c_3 = 1, \quad r_3 = 2 \end{aligned}$$

Suponha que desejamos aproximar o menor autovalor; como o disco d_3 é aquele que se encontra mais à esquerda em comparação aos demais, podemos utilizar o seu centro como fator de translação. Então, utilizamos o método da iteração inversa com $\sigma = 1$, vetor inicial $z_0 = \sqrt{3}^{-1}(1,1,1)^T$ e tolerância 10^{-5} e obtemos $\lambda = 0,5764$ e $z = (-0,0597, -0,3380, 0,9393)^T$ após 10 iterações. Por outro lado, se tivéssemos utilizado $\sigma = c_3 + r_3$, a convergência para o mesmo autovalor seria obtida em apenas 4 iterações.

6.5.4 O método da iteração inversa e o quociente de Rayleigh

Como visto no teorema 6.2.2, o valor do autovalor dominante λ_1 de uma matriz real simétrica é o máximo do quociente de Rayleigh, dentre todos os vetores $x \neq 0$. Isso nos permite utilizar a expressão (6.12) juntamente com o método da iteração inversa, conforme mostra o algoritmo 6.5.4.

Algoritmo 6.5.4 *Método da iteração inversa com translação via quociente de Rayleigh)*

proc iteração_inversa_translação(input: A, z_0, ϵ, $k_{\max}$; output: λ_k, z_k)
  $z_0 \leftarrow z_0/\| z_0 \|$
  for $k = 0, 1, \ldots, k_{\max}$ *do*
    $\lambda_k = \frac{z_k^T A z_k}{z_k^T z_k}$
    Resolva o sistema $(A - \lambda_k I)q = z_k$
    $z_k \leftarrow q/\| q \|$
    if $\| z_k - z_{k-1} \| < \epsilon$ *then*
      break
    endif
  endfor
endproc

Note que, no algoritmo 6.5.4, um sistema de equações diferente é resolvido a cada iteração, já que uma nova estimativa λ_k é utilizada a cada iteração. É possível, no entanto, que o sistema $A - \lambda_k I$ seja singular e, nesse caso, o processo deve ser terminado.

6.6 Exercícios

Exercício 6.1 *Determine os autovalores e autovetores correspondentes da matriz*

$$\begin{bmatrix} 1 & -1 & -1 \\ 0 & 2 & 5 \\ 0 & 0 & -1 \end{bmatrix}.$$

Exercício 6.2 *Calcule o autovalor dominante de*

$$\begin{bmatrix} 10 & 9 & 8 \\ 3 & 5 & 6 \\ 7 & 2 & -1 \end{bmatrix}.$$

Exercício 6.3 *Calcule o autovalor dominante e o autovetor correspondente da matriz*

$$\begin{bmatrix} 6 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

usando o método da potência, com $z_0 = (1,1,1)^T$ e tolerância 10^{-5}.

Exercício 6.4 *Calcule o autovalor dominante e o autovetor correspondente da matriz*

$$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix},$$

usando o método da potência, com $z_0 = (1,1,1)^T$ *e tolerância* 10^{-5}.

Exercício 6.5 *Explique o que acontece com o método da potência para a matriz*

$$\begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix},$$

com $z_0 = (1,1,1)^T$ *e tolerância* 10^{-5}, *sabendo que os seus autovalores são* 1, 2 *e* 3. *Repita para* $z_0 = (1,0,10^{-6})^T$.

Exercício 6.6 *Utilize o método da iteração inversa para calcular o menor autovalor da matriz*

$$\begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix},$$

com $z_0 = (1,1,1)^T$ *e tolerância* 10^{-5}.

Exercício 6.7 *Seja a matriz*

$$A = \begin{bmatrix} 5 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Explique o que ocorre com o método da iteração inversa utilizado com $\sigma = 3$, $z_0 = (1,0,0)^T$ *e tolerância* 10^{-5}. *Generalize a sua resposta.*

Capítulo 7

Interpolação

7.1 Introdução

Freqüentemente, deparamo-nos com um conjunto discreto de valores de uma função que podem ser dados na forma de tabela ou de um conjunto de medidas. Estes valores, na verdade, representam um conjunto de pontos pertencentes a uma função contínua.

Exemplo 7.1 *A seguinte tabela relaciona o calor específico (c) da água e a temperatura (T) em* ${}^0C$*:*

T	20	25	30	35	40	45	50
c	0,99907	0,99852	0,99826	0,99818	0,99828	0,99849	0,99878

Suponha que se queira calcular o calor específico da água a 32^0C *ou a temperatura para a qual o calor específico é* $0,99837$*. A interpolação ajuda a resolver este tipo de problema, já que a informação desejada não se encontra disponível na tabela.*

Seja, então, uma tabela da forma correspondente aos valores de uma função f em $n+1$ pontos

x	x_0	x_1	x_2	...	x_n
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	...	$f(x_n)$

Tabela 7.1: Valores de f em pontos distintos

reais distintos $x_0, x_1, \ldots, x_n$. Seja x^* um ponto distinto dos pontos x_i da tabela, pertencente ao intervalo que contém os pontos x_i, isto é, $x^* \neq x_i$, para $i = 0, 1, \ldots, n$; considera ainda que existem k e j, $0 \leq k \neq j \leq n$ tais que $x_k < x^* < x_j$. Então, podemos dizer que *interpolar* o ponto x^* à tabela 7.1 significa calcular o valor de $f(x^*)$, ou seja, incluir o ponto $(x^*, f(x^*))$ à tabela 7.1.

A necessidade de se efetuar esta substituição surge em várias situações, como por exemplo:

- quando são conhecidos os valores numéricos da função para um conjunto de pontos e é necessário calcular o valor da função em um ponto não tabelado (como no exemplo 7.1);
- quando a função em estudo tem uma expressão tal que operações como diferenciação e integração são difíceis de serem realizadas.

Exemplo 7.2 *Considere o problema de determinar o seno de* $6,5$ *graus. Assuma a disponibilidade de uma tabela de senos na qual os valores são dados em intervalos de* 1 *grau. Para determinar o valor desejado, tem-se três escolhas:*

Uma característica importante da interpolação polinomial é que o polinômio interpolador é *único*:

Teorema 7.2.1 *Unicidade do polinômio interpolador: Se $x_0, x_1, \ldots, x_n$ são números reais, distintos, então para números arbitrários $y_0, y_1, \ldots, y_n$, existe um polinômio único p_n, de grau máximo n, tal que $p_n(x_i) = y_i$, $0 \leq i \leq n$.*
Prova:

Unicidade: *suponha dois polinômios p_n e q_n; então o polinômio $p_n - q_n$ tem a propriedade $(p_n - q_n)(x_i) = 0$ para $0 \leq i \leq n$. Como o grau de $p_n - q_n$ é no máximo n, esse polinômio pode ter no máximo n raízes se ele não é o polinômio nulo. Como, por hipótese, os x_i são distintos, $p_n - q_n$ tem $n+1$ zeros – logo ele deve ser nulo, de onde $p_n \equiv q_n$.*

Existência: *(por indução) Para $n = 0$, obviamente existe uma função constante p_0 (de grau 0) que pode ser escolhida tal que $p_0(x_0) = y_0$.*

Suponha, agora, que tenhamos obtido um polinômio p_{k-1} de grau menor ou igual a $k-1$, tal que $p_{k-1}(x_i) = y_i$, $0 \leq i \leq k-1$. A partir desse, queremos construir um p_k na forma

$$p_k(x) = p_{k-1}(x) + c(x - x_0)(x - x_1)\ldots(x - x_{k-1})$$

o qual é um polinômio de grau k. Além disso, p_k interpola os pontos que p_{k-1} interpola, pois $p_k(x_i) = p_{k-1}(x_i) = y_i$, para $0 \leq i \leq k-1$.

Para determinarmos o coeficiente c, fazemos $p_k(x_k) = y_k$, de onde

$$y_k = p_{k-1}(x) + c(x - x_0)(x - x_1)\ldots(x - x_{k-1})$$

a qual apresenta solução única pois os termos multiplicadores de c não são nulos. $\Diamond$

Cabe salientar que, por questões de estabilidade numérica, não é adequado resolver-se o sistema (7.1), uma vez que a inversa da matriz de Vandermonde poderá ser altamente mal-condicionada, dependendo dos valores de x_i e $f(x_i)$. O exemplo a seguir mostra alguns dos problemas envolvidos na determinação do polinômio interpolador utilizando o sistema (7.1).

Exemplo 7.4 *Obtenha $p_3(x)$ que interpola $f(x)$ nos pontos x_0, x_1, x_2 e x_3 de acordo com a tabela abaixo:*

x	0,1	0,2	0,3	0,4
$f(x)$	5	13	−4	−8

O sistema linear resultante para esta tabela é

$$\begin{aligned} a_0 + 0,1\,a_1 + 0,01\,a_2 + 0,001\,a_3 &= 5 \\ a_0 + 0,2\,a_1 + 0,04\,a_2 + 0,008\,a_3 &= 13 \\ a_0 + 0,3\,a_1 + 0,09\,a_2 + 0,027\,a_3 &= -4 \\ a_0 + 0,4\,a_1 + 0,16\,a_2 + 0,064\,a_3 &= -8 \end{aligned}$$

Usando aritmética de ponto flutuante com três dígitos e o método de eliminação de Gauss, o resultado é

$$p_3(x) = -0,66 \times 10^2 + (0,115 \times 10^4)\,x - (0,505 \times 10^4)\,x^2 + (0,633 \times 10^4)\,x^3$$

e, para $x = 0,4$, obtém-se

$$p_3(x) = -9 \neq -8 = f(0,4)$$

o que obviamente está errado.

Esse exemplo mostra que nem sempre se pode utilizar o sistema (7.1) para determinar o polinômio interpolador e, usualmente, utilizam-se outras técnicas, como as formas de *Newton* e de *Lagrange*, dentre outras.

1. *Usar série de Taylor para calcular o seno com uma certa exatidão pré-definida;*
2. *Tentar encontrar uma tabela que liste o valor do seno em intervalos menores e procurar o valor exato;*
3. *Usar os senos de* 6 *e* 7 *dados na tabela disponível para tentar determinar o seno de* $6,5$, *ou seja, realizar uma interpolação.*

A função interpoladora pode ser de diversos tipos: *polinomial, exponencial,* entre outras. Veremos, a seguir, como estabelecer o *polinômio interpolador* de um conjunto de pontos.

7.2 Interpolação polinomial

Dados os pontos $(x_0, f(x_0))$, $(x_1, f(x_1))$, ..., $(x_n, f(x_n))$, portanto $n+1$ pontos, deseja-se interpolar $f(x)$ por um polinômio $p_n(x)$, de grau menor ou igual a n, tal que

$$f(x_k) = p_n(x_k) \quad \text{para} \quad k = 0, 1, \ldots, n$$

Representa-se $p_n(x)$ por $p_n(x) = a_0 + a_1 x + \ldots + a_n x^n$. Portanto, obter $p_n(x)$ significa obter os coeficientes $a_0, a_1, \ldots, a_n$. Da condição $f(x_k) = p_n(x_k)$ para $k = 0, 1, \ldots, n$, monta-se o seguinte sistema linear:

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_n x_0^n &= f(x_0) \\ a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_n x_1^n &= f(x_1) \\ \qquad\qquad\qquad\qquad\qquad\quad \vdots & \\ a_0 + a_1 x_n + a_2 x_n^2 + \ldots + a_n x_n^n &= f(x_n) \end{cases} \tag{7.1}$$

com $n+1$ equações e $n+1$ variáveis: $a_0, a_1, \ldots, a_n$.

A matriz dos coeficientes é uma *matriz de Vandermonde,*

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix} \tag{7.2}$$

a qual admite inversa desde que os pontos $x_0, x_1, \ldots, x_n$ sejam distintos.

Exemplo 7.3 *Encontre o polinômio de grau menor ou igual a* 2 *que interpola os pontos da tabela:*

x	-1	0	2
$f(x)$	4	1	-1

Solução: *Tem-se que* $p_2(x) = a_0 + a_1 x + a_2 x^2$ *e portanto,*

$$\begin{array}{lll} p_2(x_0) = f(x_0) & \Rightarrow & a_0 - a_1 + a_2 = 4 \\ p_2(x_1) = f(x_1) & \Rightarrow & a_0 = 1 \\ p_2(x_2) = f(x_2) & \Rightarrow & a_0 + 2\,a_1 + 4\,a_2 = -1 \end{array}$$

Resolvendo o sistema linear, obtém-se $a_0 = 1$, $a_1 = -\frac{7}{3}$, $a_2 = \frac{2}{3}$. *Assim,*

$$p_2(x) = 1 - \frac{7}{3}x + \frac{2}{3}x^2$$

é o polinômio que interpola $f(x)$ *em* $x_0 = -1$, $x_1 = 0$ *e* $x_2 = 2$.

Algoritmo 7.3.1 *Polinômio interpolador de Newton*

```
proc polinômio_interpolador_de_Newton(input: n, [x_0, x_1, ..., x_{n-1}], [y_0, y_1, ..., y_{n-1}];
    output: [c_0, c_1, ..., c_n])
c_0 ← y_0
    for k = 1, 2, ..., n do
        d ← x_k − x_{k-1}
        u ← c_{k-1}
        for i = k − 2, k − 3, ..., 0 do
            u ← (x_k − x_i)u + c_i
            d ← d(x_k − x_i)
        endfor
        c_k ← (y_k − u)/d
    endfor
endproc
```

Vejamos como utilizar as equações e algoritmos visto nessa seção:

Exemplo 7.5 *Dados os pontos tabulados abaixo, obtenha o polinômio interpolador na forma de Newton:*

i	0	1	2	3
x	5	-7	-6	0
y	1	-23	-54	-954

Solução:

$$\begin{aligned} c_0 &\leftarrow y_0 = 1 \\ c_1 &\leftarrow \frac{y_1 - u}{d} = \frac{-23-1}{-12} = 2 \\ c_2 &\leftarrow \frac{y_2 - u}{d} = \frac{-54+21}{-11} = 3 \\ c_3 &\leftarrow \frac{y_3 - u}{d} = \frac{-954+114}{-210} = 4 \end{aligned}$$

de onde o polinômio interpolador pode ser escrito como

$$p_3(x) = 1 + 2(x-5) + 3(x-5)(x+7) + 4(x-5)(x+7)(x+6).$$

É fácil verificar, por inspeção, que o polinômio $p_3(x)$ satisfaz $p_3(5) = 1$; para os demais pontos tabulados, basta avaliar $p_3(x)$ em cada um deles.

O polinômio interpolador, bem como os pontos interpolados, é mostrado na figura 7.1.

7.4 Forma de Lagrange

Podemos obter o polinômio único que interpola um dado conjunto de pontos utilizando a forma de Lagrange, a qual expressa $p(x)$ como

$$p(x) = y_0 l_0(x) + y_1 l_1(x) + \ldots + y_n l_n(x) = \sum_{k=0}^{n} y_k l_k(x) \tag{7.5}$$

onde $l_i(x)$ são polinômios dependentes apenas de $x_0, x_1, \ldots, x_n$ (e não de y_i).

O polinômio l_0 é da forma

$$l_0(x) = c(x-x_1)(x-x_2)\ldots(x-x_n) = c\prod_{j=1}^{n}(x-x_j)$$

7.3 Forma de Newton

Note que o processo de determinação de p_k, na prova do teorema 7.2.1, é recursivo. Além disso, p_k é obtido a partir de p_{k-1} pela adição de um único termo; logo, ao fim do processo, p_k será uma soma de termos, de tal forma que cada $p_0, p_1, \ldots, p_{k-1}$ será identificável em p_k.

O polinômio p_k tem a forma

$$\begin{aligned} p_k(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \ldots + c_k(x - x_0)\ldots(x - x_{k-1}) = \\ &= \sum_{i=0}^{k} c_i \prod_{j=0}^{i-1}(x - x_j) \end{aligned} \tag{7.3}$$

onde $\left(\prod_{j=0}^{m}(x - x_j)\right) = 1$ quando $m < 0$.

Procedendo ao uso da fórmula (7.3), os primeiros *polinômios interpoladores na forma de Newton* são os seguintes:

$$\begin{aligned} p_0(x) &= c_0 \\ p_1(x) &= c_0 + c_1(x - x_0) \\ p_2(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) \end{aligned}$$

Por questões de eficiência e de estabilidade numérica, os polinômios acima são avaliados utilizando *multiplicação aninhada*, também conhecida como *fórmula de Horner*. Suponha o polinômio interpolador na forma de Newton escrito como

$$u = \sum_{i=0}^{k} c_i \prod_{j=0}^{i-1} d_j = c_0 + c_1 d_0 + c_2 d_0 d_1 + \ldots + c_k d_0 d_1 \ldots d_{k-1};$$

podemos, então, reescrevê-lo na forma aninhada

$$u = (\ldots(((c_k)d_{k-1} + c_{k-1})\, d_{k-2} + c_{k-2})\, d_{k-3} + \ldots + c_1)\, d_0 + c_0$$

a qual pode ser calculada partindo do parênteses mais interno:

$$\begin{aligned} u_k &\leftarrow c_k \\ u_{k-1} &\leftarrow u_k d_{k-1} + c_{k-1} \\ u_{k-2} &\leftarrow u_{k-1} d_{k-2} + c_{k-2} \\ &\ddots \\ u_0 &\leftarrow u_1 d_0 + c_0 \end{aligned}$$

e, como u_0 contém o valor de u, basta usar o seguinte algoritmo, já substituindo d_k por $(x - x_k)$, conforme aparece na Equação (7.3):

$u \leftarrow c_k$
<u>for</u> $i = k-1, k-2, \ldots, 0$ <u>do</u>
  $u \leftarrow (x - x_i)u + c_i$
<u>endfor</u>

de forma a se calcular $u \equiv p_k(x)$.

Resta-nos, agora, calcular os coeficientes c_k do polinômio p_n; c_k é dado por

$$c_k = \frac{y_k - p_{k-1}(x_k)}{(x_k - x_0)(x_k - x_1)\ldots(x_k - x_{k-1})} \tag{7.4}$$

com a qual podemos escrever o algoritmo 7.3.1:

Exemplo 7.7 *Dados os pontos tabulados, obtenha o polinômio interpolador na forma de Lagrange:*

i	0	1	2	3
x	5	-7	-6	0
y	1	-23	-54	-954

Solução:

$$\begin{aligned} l_0(x) &= \frac{(x+7)(x+6)(x)}{(5+7)(5+6)(5)} \\ l_1(x) &= \frac{(x-5)(x+6)(x)}{(-7-5)(-7+6)(-7)} \\ l_2(x) &= \frac{(x-5)(x+7)(x)}{(-6-5)(-6+7)(-6)} \\ l_3(x) &= \frac{(x-5)(x+7)(x+6)}{(0-5)(0+7)(0+6)} \end{aligned}$$

de onde o polinômio interpolador pode ser escrito como

$$\begin{aligned} p_3(x) = \; & 1\left(\tfrac{(x+7)(x+6)(x)}{660}\right) - 23\left(\tfrac{(x-5)(x+6)(x)}{-84}\right) \\ & -54\left(\tfrac{(x-5)(x+7)(x)}{66}\right) - 954\left(\tfrac{(x-5)(x+7)(x+6)}{-210}\right) \end{aligned}$$

Note como esse polinômio tem uma forma bastante diferente da do polinômio interpolador de Newton; no entanto, ele igualmente satisfaz à condição de interpolação, o que pode ser verificado avaliando-se $p_3(x)$ em cada ponto. Particularmente, $p_3(5) = 1\frac{12\cdot 11\cdot 5}{660} = 1$.

7.5 Forma de Newton com diferenças divididas

Seja f uma função calculável em pontos (ou *nós*) $x_0, x_1, \ldots, x_n$, distintos mas não necessariamente ordenados. Como sabemos, existe um único polinômio p, de grau n no máximo, que interpola f nos $n+1$ nós:

$$p(x_i) = f(x_i), \quad 0 \le i \le n$$

Evidentemente, p pode ser construído a partir das bases 1, x, x^2, ..., x^n, mas isso não é recomendado devido a problemas de instabilidade numérica. Portanto, utilizam-se bases mais amenas, como aquelas do polinômio interpolador na forma de Newton:

$$\begin{aligned} q_0(x) &= \\ q_1(x) &= (x - x_0) \\ q_2(x) &= (x - x_0)(x - x_1) \\ &\ddots \\ q_n(x) &= (x - x_0)(x - x_1)\ldots(x - x_{n-1}) \end{aligned}$$

o que leva à já conhecida forma de Newton,

$$p(x) = \sum_{j=0}^{n} c_j q_j(x).$$

Utilizando a condição de interpolação ($p(x_i) = f(x_i)$), obtemos um sistema de equações para a determinação dos coeficientes c_j:

$$\sum_{j=0}^{n} c_j q_j(x_i) = f(x_i), \quad 0 \le i \le n$$

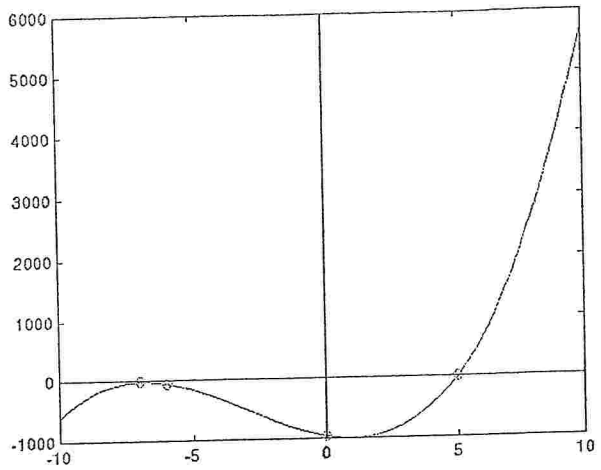

Figura 7.1: A curva do polinômio interpolador de Newton, $p_3(x) = 1 + 2(x-5) + 3(x-5)(x+7) + 4(x-5)(x+7)(x+6)$, e os pontos interpolados.

onde o coeficiente c é obtido substituindo x por x_0 na equação acima, de onde, assumindo que $y_0 = 1$,

$$1 = c\prod_{j=1}^{n}(x_0 - x_j) \rightarrow c = \prod_{j=1}^{n}(x_0 - x_j)^{-1}$$

logo,

$$l_0(x) = \prod_{j=1}^{n}\left(\frac{(x-x_j)}{(x_0-x_j)}\right);$$

os demais $l_i(x)$ são obtidos de forma similar, podendo ser expressos por

$$l_i(x) = \prod_{j=1}^{n}\left(\frac{(x-x_j)}{(x_i-x_j)}\right). \tag{7.6}$$

Os polinômios $l_i(x)$ são também conhecidos como *funções cardinais*. O exemplo que segue mostra como obter o polinômio interpolador na forma de Lagrange.

Exemplo 7.6 *A fórmula de Lagrange para a interpolação de dois pontos distintos* $(x_0, f(x_0))$ *e* $(x_1, f(x_1))$ *é:*

$$p_1(x) = y_0 l_0(x) + y_1 l_1(x)$$

onde

$$l_0(x) = \frac{(x-x_1)}{(x_0-x_1)}, \qquad l_1(x) = \frac{(x-x_0)}{(x_1-x_0)}$$

Assim,

$$p_1(x) = y_0\frac{(x-x_1)}{(x_0-x_1)} + y_1\frac{(x-x_0)}{(x_1-x_0)}$$

ou seja,

$$p_1(x) = \frac{(x_1-x)y_0 + (x-x_0)y_1}{x_1-x_0}$$

que é exatamente a equação da reta que passa por $(x_0, f(x_0))$ *e* $(x_1, f(x_1))$.

O polinômio interpolador na forma de Newton pode, agora, ser escrito como

$$p(x) = \sum_{k=0}^{n} c_k q_k(x) = \sum_{k=0}^{n} f[x_0, x_1, \ldots, x_k] \prod_{j=0}^{k-1} (x - x_j) \tag{7.10}$$

o qual é equivalente à Equação (7.3), mas os coeficientes c_k são mais facilmente obtidos em termos de diferenças divididas, usando o esquema mostrado na Tabela 7.2. Note que os coeficientes do polinômio, conforme a Equação (7.10), aparecem na *primeira* linha da tabela.

x	y	Coeficientes				
x_0	$f[x_0]$	$f[x_0,x_1]$	$\rightarrow$	$f[x_0,x_1,x_2]$	$\rightarrow$	$f[x_0,x_1,x_2,x_3]$
x_1	$f[x_1]$	$f[x_1,x_2]$	$\rightarrow$	$f[x_1,x_2,x_3]$		
x_2	$f[x_2]$	$f[x_2,x_3]$				
x_3	$f[x_3]$					

Tabela 7.2: Esquema de construção dos coeficientes do polinômio interpolador por diferenças divididas: as flechas indicam as dependências.

O exemplo que segue ilustra a construção do polinômio interpolador usando diferenças divididas.

Exemplo 7.8 *Dados os pontos tabulados abaixo, obtenha o polinômio interpolador na forma de Newton:*

i	0	1	2	3
x	5	-7	-6	0
y	1	-23	-54	-954

Solução:

x	y	*Coeficientes*		
5	1	$f[x_0,x_1] = 2$	$f[x_0,x_1,x_2] = 3$	$f[x_0,x_1,x_2,x_3] = 4$
-7	-23	$f[x_1,x_2] = -31$	$f[x_1,x_2,x_3] = -17$	
-6	-54	$f[x_2,x_3] = -150$		
0	-954			

onde os coeficientes foram obtidos como segue:

$$\begin{aligned}
f[x_0,x_1] &= \frac{f[x_1]-f[x_0]}{x_1-x_0} = \frac{-23-1}{-7-5} = 2 \\
f[x_1,x_2] &= \frac{f[x_2]-f[x_1]}{x_2-x_1} = \frac{-54-(-23)}{-6-(-7)} = -31 \\
f[x_2,x_3] &= \frac{f[x_3]-f[x_2]}{x_3-x_2} = \frac{-954-(-54)}{0-(-6)} = -150 \\
f[x_0,x_1,x_2] &= \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} = \frac{-31-2}{-6-5} = 3 \\
f[x_1,x_2,x_3] &= \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1} = \frac{-150-(-31)}{0-(-7)} = -17 \\
f[x_0,x_1,x_2,x_3] &= \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0} = \frac{-17-3}{0-5} = 4
\end{aligned}$$

O polinômio interpolador pode ser escrito, então, como

$$\begin{aligned}
p_3(x) &= f[x_0] + f[x_0,x_1](x-x_0) + \\
&\quad f[x_0,x_1,x_2](x-x_0)(x-x_1) + \\
&\quad f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2) = \\
&= 1 + 2(x-5) + 3(x-5)(x+7) + 4(x-5)(x+7)(x+6)
\end{aligned}$$

Note que, se escrevermos o sistema acima na forma $Ax = b$, a matriz A, de ordem $n+1$, tem como elementos $a_{ij} = q_j(x_i)$, $0 \leq i \leq n$, $0 \leq j \leq n$. No entanto, como

$$q_j(x_i) = \prod_{k=0}^{j-1}(x_i - x_k) = 0, \quad se\ i \leq j-1$$

então A é uma matriz triangular inferior. Por exemplo, com três nós x_0, x_1, x_2, temos

$$\begin{aligned} p_2(x) &= c_0 q_0 + c_1 q_1 + c_2 q_2 \\ &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) \end{aligned}$$

e, para determinarmos os c_i, fazemos $p_2(x_0) = f(x_0)$, $p_2(x_1) = f(x_1)$ e $p_2(x_2) = f(x_2)$, de onde obtemos o sistema

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & (x_1 - x_0) & 0 \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{bmatrix} \tag{7.7}$$

Analisando a estrutura do sistema (7.7), vemos que é possível obter os coeficientes c_i em ordem crescente, i.e. c_0, c_1, ...; além disso, o coeficiente c_i depende apenas dos valores de $f(x_0)$, $f(x_i)$, ..., $f(x_i)$, e essa dependência é denotada por

$$c_n = f[x_0, x_1, \ldots, x_n]. \tag{7.8}$$

onde c_n é o coeficiente de q_n quando $\sum_{k=0}^{n} c_k q_k$ interpola f em $x_0, x_1, \ldots, x_n$. Como

$$q_n = (x - x_0)(x - x_1)\ldots(x - x_{n-1}) = x^n + O(n-1)$$

podemos igualmente dizer que $f[x_0, x_1, \ldots, x_n]$ é o coeficiente de x^n. À expressão $f[x_0, x_1, \ldots, x_n]$ chamamos de *diferenças divididas*, pois ela tem a forma de uma divisão de duas subtrações, conforme veremos a seguir.

Algumas das fórmulas de diferenças divididas são as seguintes:

1. $f[x_0]$ é o coeficiente de x^0 no polinômio de grau 0 que interpola f em x_0; logo, $f[x_0] = f(x_0)$.

2. $f[x_0, x_1]$ é o coeficiente de x^1 no polinômio de grau 1, no máximo, interpolando f em x_0 e x_1. Como esse polinômio é

$$p_x = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

vemos que

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

3. Para diferenças divididas de maior ordem, podemos nos valer de um teorema que nos diz que elas satisfazem a relação

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0} \tag{7.9}$$

de onde podemos obter, por exemplo:

$$\begin{aligned} f[x_0, x_1] &= \frac{f[x_1] - f[x_0]}{x_1 - x_0} \\ f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ f[x_0, x_1, x_2, x_3] &= \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} \end{aligned}$$

a qual é bastante similar àquela utilizada para a determinação do polinômio interpolador de Newton através de diferença-divididas. É possível estabelecer-se a seguinte relação entre diferenças divididas e simples:

$$f[x_{i+k},\ldots,x_i] = \frac{\Delta^k y_i}{k!\,h^k} \tag{7.12}$$

para h constante.

Efetuando-se a mudança de variável

$$z = \frac{x - x_0}{h}, \tag{7.13}$$

o polinômio interpolador de Newton com diferenças simples (7.11) pode ser escrito como

$$\begin{aligned} p_n(z) &= y_0 + z\,\Delta y_0 + \frac{z(z-1)}{2!\,h^2}\Delta^2 y_0 + \ldots \\ &\quad + \frac{z(z-1)\ldots(z-n+1)}{n!\,h^n}\Delta^n y_0 \end{aligned} \tag{7.14}$$

Exemplo 7.9 *Dada a tabela*

x_i	1	2	3	4
y_i	1	9	25	55

correspondente aos valores de uma função f, interpole o ponto $x = 2,5$.

Solução: *Como os pontos são eqüidistantes, pode-se construir a tabela*

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
0	1	1	8	8	6
1	2	9	16	14	
2	3	25	30		
3	4	55			

Com isto, a fórmula para o polinômio interpolador de Newton com diferenças simples (7.11) é

$$\begin{aligned} p_3(x) &= y_0 + \frac{x - x_0}{h}\Delta y_0 + \frac{(x-x_0)(x-x_1)}{2!\,h^2}\Delta^2 y_0 \\ &\quad + \frac{(x-x_0)(x-x_1)(x-x_2)}{3!\,h^3}\Delta^3 y_0 \end{aligned}$$

ou,

$$p_3(x) = 1 + \frac{x-1}{1}8 + \frac{(x-1)(x-2)}{2!\,1}8 + \frac{(x-1)(x-2)(x-3)}{3!\,1}6$$

de onde $p_3(2,5) = 15,625$.

7.7 Interpolação inversa

O problema da interpolação inversa consiste em, dado $\bar{y} \in (f(x_0), f(x_n))$, obter $\bar{x}$ tal que $f(\bar{x}) = \bar{y}$. Este problema pode ser resolvido de duas formas:

1. Obter $p_n(x)$ que interpola $f(x)$ em $x_0, x_1, \ldots, x_n$ e em seguida encontrar $\bar{x}$ tal que $p_n(\bar{x}) = \bar{y}$;
2. Se $f(x)$ for inversível num intervalo contendo $\bar{y}$, fazer a interpolação de $\bar{x} = f^{-1}(\bar{y}) = g(\bar{y})$. Uma condição para que uma função contínua num intervalo $[a, b]$ seja inversível é que ela seja monótona crescente ou decrescente neste intervalo. Basta então considerar x como função de y e aplicar um método de interpolação conhecido: $x = f^{-1}(y) = g(y) \approx p_n(y)$.

Exemplo 7.10 *Dada a tabela*

o qual é idêntico ao polinômio interpolador de Newton mostrado no exemplo na seção 7.3; porém, com o esquema de diferenças divididas, ele é facilmente obtido.

Os coeficientes do polinômio interpolador, calculados por diferenças divididas, podem ser obtidos, também, através do seguinte algoritmo:

Algoritmo 7.5.1 *Diferenças-divididas*

proc diferenças_divididas(input: n, $[x_0, x_1, \ldots, x_{n-1}]$, $[y_0, y_1, \ldots, y_{n-1}]$;
output: $[c_0, c_1, \ldots, c_n]$)
  for $j = 0, 1, \ldots, n$ *do*
    $c_{j,0} = y_j$
  endfor
  for $j = 1, 2, \ldots, n$ *do*
    for $i = 0, 1, \ldots, j-1$ *do*
      $c_{i,j} \leftarrow \frac{c_{i+1,j-1} - c_{i,j-1}}{x_{i+j} - x_i}$
    endfor
  endfor
endproc

onde os $c_{i,j}$ são os coeficientes desejados; particularmente, $c_{0,0} = f[x_0]$, $c_{0,1} = f[x_0, x_1]$, $c_{1,1} = f[x_1, x_2]$, e assim por diante.

7.6 Forma de Newton com diferenças simples

Sejam os valores $y = f(x)$ dados através da tabela (x_i, y_i), para $i = 0, 1, \ldots, n$, onde os valores de x são eqüidistantes, isto é, $x_{i+1} - x_i = h$. Assim define-se o polinômio

$$\begin{aligned} p_n(x) &= y_0 + \frac{x - x_0}{h}\Delta y_0 + \frac{(x - x_0)(x - x_1)}{2!\,h^2}\Delta^2 y_0 + \ldots \\ &\quad + \frac{(x - x_0)(x - x_1)\ldots(x - x_{n-1})}{n!\,h^n}\Delta^n y_0 \end{aligned} \tag{7.11}$$

onde $\Delta^k y_i$ é uma *diferença simples* de ordem k, calculada conforme

$$\Delta y_i = y_{i+1} - y_i \qquad \text{Ordem 1}$$

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i = y_{i+2} - 2y_{i+1} + y_i \qquad \text{Ordem 2}$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i \qquad \text{Ordem } k$$

e, a fim de facilitar a determinação dos valores $\Delta^k y_i$, pode-se construir uma tabela como a que segue:

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$
0	x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$
1	x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	
2	x_2	y_2	Δy_2	$\Delta^2 y_2$		
3	x_3	y_3	Δy_3			
4	x_4	y_4				

Em cada intervalo $[t_i, t_{i+1}]$, S é dada por um polinômio cúbico diferente.

Seja S_i o polinômio cúbico que representa S no intervalo $[t_i, t_{i+1}]$; então, como $S = S_0(x)|t_0 \leq x \leq t_1$ e $S = S_1(x)|t_1 \leq x \leq t_2$, temos necessariamente, em $x = t_1$, que $S_0(t_1) = y_1 = S_1(t_1)$. Generalizando, exigimos que $S_{i-1}(t_i) = y_i = S_i(t_i)$, $1 \leq i \leq n-1$. Além disso, exigimos que S' e S'' sejam contínuas, a fim de podermos impor as condições necessárias para que S seja contínua ao longo do intervalo $[t_0, t_n]$.

Vejamos se a continuidade de S, S' e S'' é suficiente para se definir a "spline" cúbica. Um polinômio cúbico tem a forma

$$a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

e, como desejamos obter um polinômio cúbico em cada um dos n intervalos, vemos que será necessário determinar $4n$ coeficientes. Em cada intervalo $[t_i, t_{i+1}]$, temos duas condições de interpolação

$$S(t_i) = y_i \tag{7.15}$$
$$S(t_{i+1}) = y_{i+1} \tag{7.16}$$

ou seja, poderemos determinar $2n$ coeficientes. A continuidade de S apenas nos dá a condição $S_{i-1}(t_i) = S_i(t_i)$, equivalente às duas condições de interpolação em cada intervalo.

Impondo continuidade de S' e de S'', podemos escrever

$$S'_{i-1}(t_i) = S'_i(t_i) \tag{7.17}$$
$$S''_{i-1}(t_i) = S''_i(t_i) \tag{7.18}$$

obtendo $2(n-1)$ outras condições. Ao todo, temos, portanto, $4n-2$ condições; as duas que faltam podem ser obtidas de várias formas mas, usualmente, escreve-se

$$S''_0(t_0) = 0 \tag{7.19}$$
$$S''_{n-1}(t_n) = 0 \tag{7.20}$$

e, com isso, temos todas as $4n$ condições necessárias para se determinar os coeficientes da "spline" cúbica.

Vejamos como definir S_i em $[t_i, t_{i+1}]$. Escrevendo $z_i = S''_i(t_i)$, temos que, por definição, z_i existe para $0 \leq i \leq n-1$ e satisfaz

$$\lim_{x \to t_i^-} S''_i(x) = z_i = \lim_{x \to t_i^+} S''_i(x), \quad 1 \leq i \leq n-1$$

pois S'' é contínua em cada nó interno (diferentes de t_0 e t_n). Como S_i é um polinômio cúbico em $[t_i, t_{i+1}]$, S''_i é uma função linear satisfazendo $S''_i(t_i) = z_i$ e $S''_i(t_{i+1}) = z_{i+1}$ e, portanto, é uma reta que passa por z_i e z_{i+1}:

$$S''_i(x) = \frac{z_i}{h_i}(t_{i+1} - x) + \frac{z_{i+1}}{h_i}(x - t_i) \tag{7.21}$$

onde $h_i = t_{i+1} - t_i$. Integrando a equação acima duas vezes, obtemos

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - t_i)^3 + C(x - t_i) + D(t_{i+1} - x) \tag{7.22}$$

onde C e D são constantes de integração. Agora, usando $S_i(t_i) = y_i$ e $S_i(t_{i+1}) = y_{i+1}$, podemos determinar C e D:

$$\begin{aligned} S_i(t_i) &= y_i \\ \frac{z_i}{6h_i}(t_{i+1} - t_i)^3 + \frac{z_{i+1}}{6h_i}(t_i - t_i)^3 + C(t_i - t_i) + D(t_{i+1} - t_i) &= y_i \\ \frac{z_i}{6}(t_{i+1} - t_i)^2 + Dh_i &= y_i \end{aligned}$$

x	0,2	0,3	0,4
$y = e^x$	1,2214	1,3499	1,4918

obter x tal que $e^x = 1,3165$ usando um processo de interpolação quadrática.

Solução:*Pode-se usar a fórmula de Newton para obter $p_2(y)$ que interpola $g(y) = f^{-1}(y)$. Para isto, constrói-se a tabela de diferenças divididas:*

i	y_i	$g[y_i]$	$g[y_{i+1}, y_i]$	$g[y_{i+2}, y_{i+1}, y_i]$
0	1,2214	0,2	0,7782	−0,2718
1	1,3499	0,3	0,7047	
2	1,4918	0,4		

O polinômio $p_2(y)$ é determinado por

$$\begin{aligned} p_2(y) &= g[y_0] + (y - y_0)\, g[y_1, y_0] + (y - y_0)(y - y_1)\, g[y_2, y_1, y_0] \\ &= 0,2 + (y - 1,2214)\, 0,7782 + (y - 1,2214)(y - 1,3499)\, (-0,2718) \end{aligned}$$

de forma que $p_2(1,3165) = 0,27487$. Assim, $e^{0,27487} = 1,3165$.

7.8 Interpolação por "splines"

Suponha que se deseja interpolar um conjunto de pontos por uma função polinomial; então, uma função "spline" consiste de um conjunto de polinômios – de grau pequeno – que atuam sobre alguns dos pontos daquele conjunto.

Formalmente, suponha que $n+1$ pontos $t_0, t_1, \ldots, t_n$, chamados de *nós*, tenham sido especificados, satisfazendo $t_0 < t_1 < \ldots < t_n$, além de um número inteiro $k \geq 0$. Uma *função "spline" de grau* k, com nós $t_0, t_1, \ldots, t_n$, é uma função S tal que

1. Em cada subintervalo $[t_{i-1}, t_i)$, S é um polinômio de grau menor ou igual a k;
2. S tem as suas primeiras $k - 1$ derivadas contínuas em $[t_0, t_n]$.

Diz-se, portanto, que S é um polinômio contínuo de grau k, por partes, apresentando derivadas de ordem igual ou inferior a $k - 1$ contínuas. Vejamos alguns exemplos:

"Spline" de grau 0: são polinômios constantes, os quais podem ser dados por

$$S(x) = \begin{cases} S_0(x) = c_0 & t_0 \leq x < t_1 \\ S_1(x) = c_1 & t_1 \leq x < t_2 \\ \vdots & \vdots \\ S_{n-1}(x) = c_{n-1} & t_{n-1} \leq x < t_n \end{cases}$$

"Spline" de grau 1: são retas que unem os nós, as quais podem ser dadas por

$$S(x) = \begin{cases} S_0(x) = a_0 x + b_0 & t_0 \leq x < t_1 \\ S_1(x) = a_1 x + b_1 & t_1 \leq x < t_2 \\ \vdots & \vdots \\ S_{n-1}(x) = a_{n-1} x + b_{n-1} & t_{n-1} \leq x < t_n \end{cases}$$

As "splines" são normalmente usadas na forma cúbica. Assuma que uma tabela de dados seja dada, i.e.,

x	t_0	t_1	$\ldots$	t_{n-1}	t_n
y	y_0	y_1	$\ldots$	y_{n-1}	y_n

Algoritmo 7.8.1 *Coeficientes da "spline"*

proc coeficientes_spline(input: n, $[t_0, t_1, \ldots, t_n]$, $[y_0, y_1, \ldots, y_n]$;
output: $[z_0, z_1, \ldots, z_n]$)
for $i = 0, 1, \ldots, n-1$ *do*
$h_i = t_{i+1} - t_i$
$b_i = \frac{6}{h_i}(y_{i+1} - y_i)$
endfor
$u_1 = 2(h_0 + h_1)$
$v_1 = b_1 - b_0$
for $i = 2, 3, \ldots, n-1$ *do*
$u_i = \frac{2(h_i + h_{i-1}) - h_{i-1}^2}{u_{i-1}}$
$v_i = \frac{b_i - b_{i-1} - h_{i-1} v_{i-1}}{u_{i-1}}$
endfor
$z_n = 0$
for $i = n-1, n-2, \ldots, 1$ *do*
$z_i = \frac{v_i - h_i z_{i+1}}{u_i}$
endfor
$z_0 = 0$
endproc

Uma vez obtidos os z_i, pode-se avaliar $S_i(x)$ usando a expressão abaixo, na forma aninhada:

$$S_i(x) = y_i + (x - t_i)\,(C_i + (x - t_i)\,(B_i - (x - t_i)A_i)) \tag{7.26}$$

onde

$$\begin{aligned} A_i &= \frac{1}{6h_i}(z_{i+1} - z_i) \\ B_i &= \frac{z_i}{2} \\ C_i &= -\frac{h_i}{6} z_{i+1} - \frac{h_i}{3} z_i + \frac{y_{i+1} - y_i}{h_i} \end{aligned}$$

Cabe relembrar que, para cada intervalo $[t_i, t_{i+1}]$, deve ser utilizada a Equação (7.26) com os valores adequados: z_i e z_{i+1}; y_i e y_{i+1}; e $h_i = t_{i+1} - t_i$.

7.9 Estudo do erro na interpolação

Seja f uma função contínua com $(n+1)$ derivadas contínuas em um intervalo I, $x_0 < x_1 < \ldots < x_n$, $(n+1)$ pontos distintos pertencentes ao intervalo I e $p_n(x)$ o polinômio interpolador de f relativamente aos pontos $x_0, x_1, \ldots, x_n$.

Ao se aproximar esta função $f(x)$ por um polinômio interpolador $p_n(x)$, de grau menor ou igual a n, comete-se um erro de truncamento na interpolação de x. Este é definido por

$$E_n(x) = f(x) - p_n(x) = (x - x_0)\,(x - x_1)\,(x - x_2) \ldots (x - x_n)\,\frac{f^{(n+1)}(\xi)}{(n+1)!} \tag{7.27}$$

para $\xi \in (x_0, x_n)$.

Para utilizar esta fórmula, é preciso conhecer $f^{(n+1)}$ e o ponto ξ. Mas, em geral, a forma analítica da função f não é conhecida, não sendo possível portanto determinar nem $f^{(n+1)}$ e, conseqüentemente, nem $E_n(x)$. Ainda assim, mesmo quando se conhece a forma analítica de f,

com $h_i = t_{i+1} - t_i$, de onde

$$D = \frac{y_i}{h_i} - \frac{z_i h_i}{6} \tag{7.23}$$

Para determinar C, escrevemos

$$\begin{aligned} S_i(t_{i+1}) &= y_{i+1} \therefore \\ \frac{z_i}{6h_i}(t_{i+1} - t_{i+1})^3 + \frac{z_{i+1}}{6h_i}(t_{i+1} - t_i)^3 + C(t_{i+1} - t_i) + D(t_{i+1} - t_{i+1}) &= y_{i+1} \\ \frac{z_{i+1}}{6}h_i^2 + Ch_i &= y_{i+1} \end{aligned}$$

de onde

$$C = \frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6} \tag{7.24}$$

Logo, podemos escrever S_i – o polinômio cúbico em cada intervalo $[t_i, t_{i+1}]$ – como

$$\begin{aligned} S_i(x) = &\frac{z_i}{6h_i}(t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6}\right)(x - t_i) + \\ &\left(\frac{y_i}{h_i} - \frac{z_i h_i}{6}\right)(t_{i+1} - x) \end{aligned}$$

Agora, resta determinar os z_i, $1 \leq i \leq n-1$, usando a continuidade em S'. Para tanto, temos $S'_{i-1}(t_i) = S'_i(t_i)$, nos nós interiores. Diferenciando a expressão (7.25) e substituindo $x = t_i$, temos

$$S'_{i-1}(t_i) = \frac{z_i h_{i-1}}{3} + \frac{z_{i-1}h_{i-1}}{6} - \frac{y_{i-1}}{h_{i-1}} + \frac{y_i}{h_i}$$

e

$$S'_i(t_i) = -\frac{z_i h_i}{3} - \frac{z_{i+1}h_i}{6} - \frac{y_i}{h_i} + \frac{y_{i+1}}{h_i}$$

de onde, igualando ambas as expressões acima, vem

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = \frac{6}{h_i}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1})$$

a qual, com $z_0 = z_n = 0$, para $1 \leq i \leq n-1$, nos leva a um sistema de equações de $n-1$ equações a $n-1$ variáveis,

$$\begin{bmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix} z = v \tag{7.25}$$

onde:

$$\begin{aligned} h_i &= t_{i+1} - t_i \\ u_i &= 2(h_i + h_{i-1}) \\ b_i &= \frac{6}{h_i}(y_{i+1} - y_i) \\ v_i &= b_i - b_{i-1} \end{aligned}$$

O sistema (7.25) pode ser resolvido por eliminação de Gauss, sem pivotamento, já que ele é diagonal dominante. Um algoritmo para resolver o referido sistema pode ser escrito como

obter $f(0,47)$ utilizando um polinômio de grau 2 e dar uma estimativa para o erro.

Solução: *A tabela de diferenças é dada por*

i	x_i	$f[x_i]$	$f[x_{i+1},x_i]$	$f[x_{i+2},x_{i+1},x_i]$	$f[x_{i+3},x_{i+2},x_{i+1},x_i]$
0	0,4	0,27	0,1667	1,0415	−2,6031
1	0,52	0,29	0,375	*0,2085*	
2	0,6	0,32	*0,4167*		
3	0,72	0,37			

Assim, usando a fórmula de Newton,

$$\begin{aligned} p_2(x) &= f[x_0] + (x-x_0)\,f[x_1,x_0] + (x-x_0)(x-x_1)\,f[x_2,x_1,x_0] \\ &= 0,27 + (x-0,4)\,0,1667 + (x-0,4)(x-0,52)\,1,0415 \end{aligned}$$

de onde $f(0,47) = p_2(0,47) = 0,2780$.

A estimativa para o erro de truncamento é

$$\begin{aligned} |E_2(x)| &\leq |(x-x_0)(x-x_1)(x-x_2)|\,|f[x_3,x_2,x_1,x_0]| \\ &\leq |(x-0,4)(x-0,52)(x-0,6)|\,|-2,6031| \end{aligned}$$

ou seja,

$$|E_2(0,47)| \leq 1,184 \times 10^{-3}.$$

7.10 Exercícios

Exercício 7.1 *Determine $f(1,32)$ a partir da tabela*

x	1,3	1,4	1,5
$f(x)$	3,669	4,055	4,482

Exercício 7.2 *Considere o exemplo 7.4. O que ocorreu naquela situação? Explique por que o valor calculado não coincide com o valor tabelado.*

Exercício 7.3 *A integral elíptica completa é definida por*

$$K(k) = \int_0^{\frac{\pi}{2}} \frac{dx}{(1-k^2\mathrm{sen}^2 k)^{\frac{1}{2}}}.$$

Consultando uma tabela de valores destas integrais, tem-se que

$$\begin{aligned} K(1) &= 1,5708 \\ K(3) &= 1,5719 \\ K(5) &= 1,5738 \end{aligned}$$

Calcule $K(3,5)$ usando um polinômio interpolador do segundo grau.

Exercício 7.4 *Interpole o ponto $x = 0{\cdot}5$ à tabela*

x	0	1	3	4
$f(x)$	−5	1	25	55

utilizando o polinômio interpolador na forma de Newton.

Exercício 7.5 *Calcule uma aproximação para $f(2,3)$ pela forma interpoladora de Lagrange.*

não se sabe o valor de ξ e, portanto, não se pode calcular $E_n(x)$ exatamente. Entretanto, pode-se delimitar o erro pela desigualdade:

$$|E_n(x)| \leq |(x-x_0)(x-x_1)(x-x_2)\ldots(x-x_n)| \max_{x\in I} \frac{|f^{(n+1)}(x)|}{(n+1)!}. \tag{7.28}$$

Exemplo 7.11 *Seja o problema de se obter* $\ln(3,7)$ *por interpolação linear, onde* $\ln(x)$ *está tabelada abaixo:*

x	1	2	3	4
$\ln(x)$	0	0,6931	1,0986	1,3863

Solução: *Como* $x = 3,7 \in (3;4)$ *escolhe-se* $x_0 = 3$ *e* $x_1 = 4$. *Pela forma de Newton, o polinômio interpolador é*

$$\begin{aligned} p_1(x) &= y_0 + \frac{x-x_0}{h}\Delta y_0 \\ &= 1,0986 + (x-3)\frac{(1,3863-1,0986)}{4-3} \\ &= 1,0986 + (x-3)(0,2877). \end{aligned}$$

Conseqüentemente, $p_1(3,7) = 1,3000$. *Neste caso, dado que, com quatro casas decimais,* $f(3,7) = \ln(3,7) = 1,3083$, *tem-se condições de calcular o erro exato:*

$$E_1(3,7) = |f(3,7) - p_n(3,7)| = |1,3083 - 1,3| = 0,0083$$

Por outro lado, tem-se também a seguinte majoração para o erro:

$$\begin{aligned} |E_1(3,7)| &\leq |(3,7-3)(3,7-4)| \max_{x\in[3;4]} \frac{|f''(x)|}{2!} \\ &\leq |(3,7-3)(3,7-4)|\frac{1}{2} \\ &\leq 0,105 \end{aligned}$$

7.9.1 Estimativa para o erro

Se a função $f(x)$ é dada em forma de tabela, o valor absoluto do erro $E_n(x)$ só pode ser estimado, já que não é possível calcular $f^{(n+1)}$. Entretanto, construindo a tabela de diferenças divididas até ordem $(n+1)$, tem-se que

$$|E_n(x)| \approx |(x-x_0)(x-x_1)\ldots(x-x_n)|\,|f[x_{n+1},x_n,\ldots,x_0]| \tag{7.29}$$

ou, para pontos igualmente espaçados,

$$|E_n(x)| \approx |(x-x_0)(x-x_1)\ldots(x-x_n)| \left|\frac{\Delta^{n+1}y_0}{(n+1)!\,h^{n+1}}\right|, \tag{7.30}$$

já que

$$f[x_{n+1},x_n,\ldots,x_0] = \frac{\Delta^{n+1}y_0}{(n+1)!\,h^{n+1}} \tag{7.31}$$

para h constante.

Exemplo 7.12 *Seja* $f(x)$ *dada na forma*

x	0,4	0,52	0,6	0,72
$f(x)$	0,27	0,29	0,32	0,37

Capítulo 8

Ajuste de dados experimentais

8.1 Introdução

Uma forma de trabalhar com uma função definida por uma tabela de valores é a interpolação polinomial. Entretanto esta não é aconselhável quando:

1. é preciso obter um valor aproximado da função em algum ponto fora do intervalo de tabelamento, ou seja, quando se quer extrapolar;
2. os valores tabelados são resultado de algum experimento físico ou de alguma pesquisa, porque, nestes casos, estes valores podem conter erros inerentes que, em geral, não são previsíveis.

Surge, então, a necessidade de se ajustar a estas funções tabeladas uma função que seja uma "boa aproximação" para os valores tabelados e que permita "extrapolar" com certa margem de segurança.

Exemplo 8.1 *Considere um teste de desempenho de um automóvel. Este é acelerado a partir do repouso e depois viaja com aceleração máxima até que sua velocidade atinja* $100km/h$. *Enquanto isto, as leituras no velocímetro são realizadas a cada* $1s$. *Quando a velocidade é graficada como função do tempo, obtém-se um conjunto de pontos. Seria esperado que estes pontos definissem uma curva suave. No entanto, erros de medida e outros fatores fazem com que os pontos não fiquem tão bem arranjados: alguns dos valores registrados para a velocidade ficam muito altos e outros, muito baixos.*

Supondo que se desejasse determinar a velocidade aos $6,5s$, *seria possível interpolar entre as leituras feitas aos* $6s$ *e* $7s$, *mas como provavelmente existe algum erro nestas medidas, o valor assim obtido poderia não ser uma boa aproximação para o valor desejado. O que fazer?*

A solução para o problema é tentar ajustar uma "provável" curva ao conjunto de dados. Como é possível que vários destes dados não sejam precisos, esta curva não precisa, necessariamente, passar por nenhum dos pontos. Por outro lado, como os erros de medida provavelmente não são tão grandes, a curva deveria pelo menos passar perto de cada ponto: provavelmente acima de uns e abaixo de outros. Na verdade, ao invés de procurar a função f que passa por cada um dos dados experimentais, calcula-se a função que melhor se ajusta a eles.

Exemplo 8.2 *Suponha que os dados abaixo – temperatura* T *a cada período de tempo* t *– foram obtidos em um experimento num laboratório:*

t	1	2	3	4	5
T	15,0	28,4	45,3	58,6	77,4

e o gráfico exibido na figura 8.2 sugere que esses dados podem ser aproximados razoavelmente bem por uma reta. Assim, se for necessário saber o valor de T *no tempo* $t = 1,5$, *podemos obter a*

x	2,0	2,4	2,6	2,8
$f(x)$	0,31495	0,020561	−0,09682	−0,18505

Exercício 7.6 *A tabela abaixo fornece a demanda diária máxima de energia elétrica em uma cidade. Encontre a data do pico máximo e o valor deste pico.*

x (data)	5 *outubro*	15 *outubro*	25 *outubro*	4 *novembro*
y (demanda)	10	15	20	13

Exercício 7.7 *A função definida pela tabela*

x	1,9	2,0	2,1	2,2	2,3
$f(x)$	−1,941	−1,000	0,061	1,248	2,567

tem uma raiz no intervalo $(2; 2,1)$. *Calcule esta raiz, aproximando a função por um polinômio de terceiro grau.*

Exercício 7.8 *Dada a tabela abaixo, calcular* $f(0,32)$ *e estimar o erro de truncamento do valor calculado.*

x	0,2	0,3	0,4	$0,\dot{5}$
$f(x)$	0,5544	0,5639	0,5735	0,5831

Exercício 7.9 *Dada a seguinte tabela para a função* $f(x) = e^x$,

x	1	1,1	1,2
e^x	2,718	3,004	3,320

calcule $f(1,05)$ *e delimite o erro para o valor interpolado, utilizando aritmética de ponto flutuante com quatro algarismos significativos.*

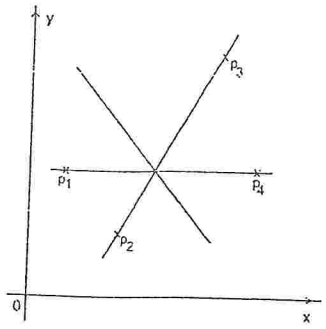

Figura 8.3: *Qual a melhor aproximação, nesse caso?*

8.2 Mínimos quadrados - domínio discreto

Para aproximar uma função $y = f(x)$ tabelada em n pontos distintos x_i, $i = 0, 1, 2, \ldots, n$, por uma função g da forma

$$\sum_{k=0}^{m} a_k \, g_k(x) \tag{8.2}$$

Precisa-se determinar $a_0, a_1, \ldots, a_m$ que minimizam a soma dos quadrados dos resíduos

$$M(a_0, a_1, \ldots, a_m)$$

nos pontos x_i, $i = 0, 1, 2, \ldots, n$. Para minimizar

$$M(a_0, a_1, \ldots, a_m) = \sum_{i=0}^{n} r_i^2(x) = \sum_{i=0}^{n} (f(x_i) - g(x_i))^2 \tag{8.3}$$

é preciso que

$$\frac{\partial M}{\partial a_0} = 0 \qquad \frac{\partial M}{\partial a_1} = 0 \quad \ldots \quad \frac{\partial M}{\partial a_m} = 0, \tag{8.4}$$

Por outro lado, certamente existem processos naturais que tem um comportamento *exponencial*, *potencial* e *quadrático*, dentro outros. É possível, para um conjunto de dados experimentais, calcular o quão boa é uma determinada aproximação, escolhida previamente. A seguir, veremos como determinar os coeficientes de uma determinada função de ajuste.

8.3 Ajuste linear

Neste caso, determina-se os parâmetros a_0 e a_1 da reta $a_0 + a_1 x$ de modo que a soma dos quadrados em cada ponto seja mínima. Em outras palavras, deseja-se determinar a_0 e a_1 que minimizem

$$M(a_0, a_1) = \sum_{i=0}^{n} r_i^2(x) = \sum_{i=0}^{n} (y_i - a_0 - a_1 x_i)^2 \tag{8.5}$$

Para isto, é necessário que

$$\frac{\partial M}{\partial a_0} = 0 \quad \text{e} \tag{8.6}$$

$$\frac{\partial M}{\partial a_1} = 0 \tag{8.7}$$

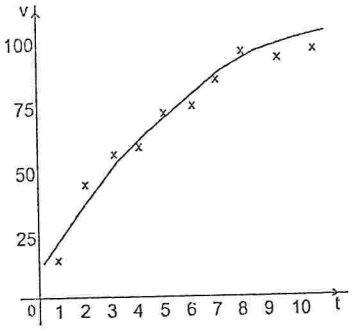

Figura 8.1: *Gráfico $v \times t$, com erros nas medidas de v.*

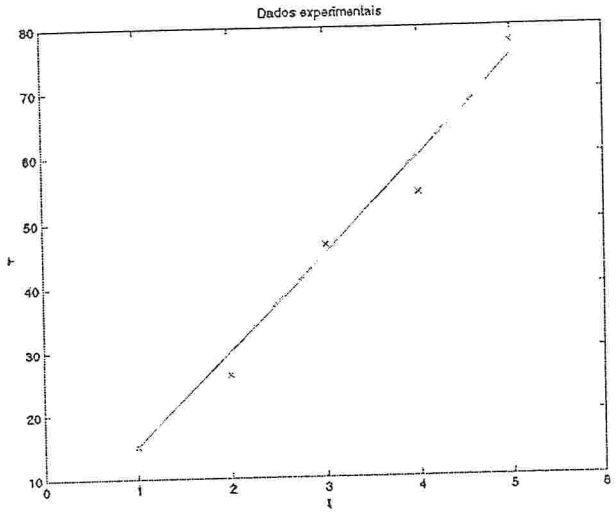

Figura 8.2: *Dados experimentais.*

equação da reta que melhor aproxima os pontos obtidos experimentalmente e, então, calcular o valor de T de acordo com aquela reta.

A pergunta que surge é: dado um conjunto de dados, como fazer o ajuste? Ao aproximar uma função f por uma função g de uma família G, é introduzido um certo erro r, denominado resíduo, isto é,

$$r(x) = f(x) - g(x) \tag{8.1}$$

Aparentemente, uma boa aproximação seria obtida fazendo $\sum_x r(x) = 0$. No entanto, isto não é verdade. Suponha que, em um certo experimento, foram obtidos os pontos p_1, p_2, p_3 e p_4. Sabendo que o fenômeno é descrito por uma reta, esta é determinada de modo a satisfazer $\sum_x r(x) = 0$. Pode-se observar na figura abaixo que as retas que foram traçadas obedecem tal critério, o que mostra que $\sum_x r(x) = 0$ não é uma boa escolha.

O problema enfrentado com este critério é o cancelamento dos erros positivos com os negativos. Uma maneira de evitar este cancelamento é trabalhar com o quadrado do resíduo e exigir que $\sum_x r^2(x)$ seja mínimo. O método para aproximar uma função f por uma $g \in G$ utilizando este último critério é denominado *método dos mínimos quadrados.*

Exemplo 8.4 *Obtenha a expressão da parábola que se ajusta aos dados da tabela:*

x	-2	-1	0	1	2	3
y	$-0,01$	$0,51$	$0,82$	$0,88$	$0,81$	$0,49$

O sistema normal para o caso de uma parábola ($p = 2$) é

$$\begin{bmatrix} 6 & 3 & 19 \\ 3 & 19 & 27 \\ 19 & 27 & 115 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 3,50 \\ 3,48 \\ 9,00 \end{bmatrix}$$

A solução deste sistema indica que a parábola que melhor se ajusta a este conjunto de dados é $g(x) = -0,102x^2 + 0,201x + 0,806$.

8.5 Ajustamento por funções não lineares nos parâmetros – linearização

O método dos mínimos quadrados pode ser empregado também aproximar uma função f por uma função g de uma família não linear nos parâmetros. Exemplos destas funções são as exponenciais, hiperbólicas e racionais, entre outras, como veremos a seguir.

8.5.1 Ajustamento por uma função exponencial

A função $y = ce^{ax}$ pode ser linearizada tomando-se o logaritmo de ambos os lados. No final, obtém-se uma relação linear entre as variáveis transformadas. O primeiro passo é

$$\ln y = \ln c + ax. \tag{8.15}$$

Agora, usando a mudança de variáveis (e de constantes)

$$Y = \ln y, \qquad X = x, \qquad a_0 = \ln c, \qquad a_1 = a \tag{8.16}$$

chega-se à relação linear entre as variáveis X e Y:

$$Y = a_1 X + a_0, \tag{8.17}$$

Sendo assim, pode-se aplicar o mesmo método utilizado para o ajustamento de uma reta aos dados transformados $\{(X_i, Y_i)\} = \{(x_i, \ln y_i)\}$. Os coeficientes a_0 e a_1 são encontrados pela solução do sistema

$$\begin{bmatrix} (n+1) & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} \ln y_i \\ \sum_{i=0}^{n} x_i \ln y_i \end{bmatrix} \tag{8.18}$$

de forma que $c = e^{a_0}$ e $a = a_1$ determinam a função de ajustamento.

Exemplo 8.5 *Ajuste os dados da tabela a uma função exponencial.*

x	0	$0,5$	1	$1,5$	2	$2,5$	3	$3,5$	4
y	3	4	6	9	12	17	24	33	48

O sistema linear para este caso é

$$\begin{bmatrix} 9 & 18 \\ 18 & 51 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 22,3378 \\ 55,0903 \end{bmatrix}$$

e a sua solução é $a_0 = 1,093337$ *e* $a_1 = 0,694319$. *Portanto,* $c = e^{a_0} = 2,984216$ *e a função exponencial procurada é* $y = 2,984216\, e^{0,694319}$.

ou seja, que

$$\frac{\partial M}{\partial a_0} = 2\sum_{i=0}^{n}(y_i - a_0 - a_1 x_i)(-1) = 0 \tag{8.8}$$

$$\frac{\partial M}{\partial a_1} = 2\sum_{i=0}^{n}(y_i - a_0 - a_1 x_i)(-x_i) = 0 \tag{8.9}$$

Organizando estas condições, tem-se

$$\begin{cases} \sum_{i=0}^{n} y_i &= \sum_{i=0}^{n} a_0 + \sum_{i=0}^{n} a_1 x_i \\ \sum_{i=0}^{n} x_i y_i &= \sum_{i=0}^{n} a_0 x_i + \sum_{i=0}^{n} a_1 x_i^2 \end{cases} \tag{8.10}$$

e chega-se ao seguinte sistema linear:

$$\begin{bmatrix} \sum_{i=0}^{n} 1 & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} x_i y_i \end{bmatrix} \tag{8.11}$$

denominado *sistema normal*. Resolvendo este sistema, são obtidos os valores de a_0 e de a_1, ou seja, determina-se a equação (reta, no caso) de ajustamento.

Exemplo 8.3 *Como resultado de algum experimento, suponha que são obtidos os seguintes valores para a função f:*

x	0	1	2	3	4
$f(x)$	0	1	1	4	4

Determine a reta que melhor se ajusta a esta função segundo o método dos mínimos quadrados.

O sistema normal correspondente é

$$\begin{bmatrix} 5 & 10 \\ 10 & 30 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 10 \\ 31 \end{bmatrix}$$

que tem solução $a_0 = -1/5$ e $a_1 = 11/10$. Portanto, a reta que aproxima $f(x)$ é

$$g(x) = \frac{11}{10}x - \frac{1}{5}$$

8.4 Ajuste polinomial

Pode-se estender o conceito de ajustamento de uma reta por mínimos quadrados para o caso geral de um polinômio de grau p. Neste caso, determina-se os parâmetros $a_0, a_1, \ldots, a_p$ do polinômio $a_0 + a_1 x + \ldots + a_p x^p$ que minimizem

$$M(a_0, a_1, \ldots, a_p) = \sum_{i=0}^{n} r_i^2(x) = \sum_{i=0}^{n}(y_i - a_0 - a_1 x_i - \ldots - a_p x^p)^2 \tag{8.12}$$

Para isto, é necessário que

$$\frac{\partial M}{\partial a_0} = \frac{\partial M}{\partial a_1} = \ldots = \frac{\partial M}{\partial a_p} = 0 \tag{8.13}$$

de onde se obtém o sistema

$$\begin{bmatrix} n+1 & \sum_{i=0}^{n} x_i & \cdots & \sum_{i=0}^{n} x_i^p \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \cdots & \sum_{i=0}^{n} x_i^{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n} x_i^p & \sum_{i=0}^{n} x_i^{p+1} & \cdots & \sum_{i=0}^{n} x_i^{2p} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} x_i y_i \\ \vdots \\ \sum_{i=0}^{n} x_i^p y_i \end{bmatrix} \tag{8.14}$$

Resolvendo este sistema, são obtidos os valores de $a_0, a_1, \ldots, a_p$, ou seja, os coeficientes do polinômio de grau p.

Desvio relativo em relação à média: Seleciona-se a regressão que tiver o *menor* desvio relativo em relação à média,

$$t = \max_{i=1}^{m} \frac{|y_i - \bar{y}|}{|\bar{y}|} \tag{8.24}$$

onde $\bar{y} = m^{-1} \sum_{i=1}^{m} y_i$.

Coeficiente de variação da amostra: Seleciona-se a regressão que apresentar o menor coeficiente de variação da amostra,

$$D = \frac{\sqrt{\frac{\sum_{i=1}^{m} (y_i - \bar{y})^2}{m-1}}}{\bar{y}} \tag{8.25}$$

É importante notar que poderá haver casos em que a escolha de uma ou outra medida favorecerá uma ou outra forma de regressão, conforme pode ser verificado nos exemplos que seguem.

Exemplo 8.6 *Dada a tabela*

x	1	2	3	4	5
f	15,0	28,4	45,3	58,6	77,4

obtenha as quatro regressões – linear, quadrática (polinomial), potencial e exponencial – calculando as medidas para escolha da melhor regressão.

Solução: *Calculadas as regressões, obtemos os seguintes valores para y_i (arredondados para a primeira casa decimal):*

x	1	2	3	4	5
linear	13,9	29,4	44,9	60,4	75,9
quadrática	15,0	28,9	43,9	59,9	77,0
potencial	14,7	29,6	44,7	59,9	75,2
exponencial	17,4	26,0	38,8	57,9	86,4

e os correspondentes valores das medidas

	e	t	D
linear	0,0707	0,6898	0,2727
quadrática	0,0307	0,7127	0,2729
potencial	0,0434	0,6771	0,2669
exponencial	0,1597	0,9079	0,3045

Analisando a tabela acima, vemos que, se o critério escolhido fosse o erro relativo e, deveríamos escolher a regressão quadrática, com coeficientes

$$a_0 = 2,04, \quad a_1 = 12,4143, \quad a_2 = 0,5143;$$

para as outras duas medidas, a regressão escolhida seria a potencial, com

$$a_0 = 14,6535, \quad a_1 = 1,0160$$

A figura (8.4) mostra que ambas as regressões, nesse caso, aproximam razoavelmente bem os dados experimentais.

Exemplo 8.7 *Suponha os dados experimentais dados por*

x	1	2	3	4	5
f	2,7183	7,3891	20,0855	54,5982	148,4132

Obtenha as quatro regressões – linear, quadrática (polinomial), potencial e exponencial – calculando as medidas para escolha da melhor regressão.

8.5.2 Ajustamento por uma função potência

A função $y = a\,x^b$ pode ser linearizada tomando-se o logaritmo: $\ln y = \ln a + b \ln x$. Com isto, mediante a mudança de variáveis $Y = \ln y$, $X = \ln x$, $a_0 = \ln a$ e $a_1 = b$, o sistema normal fica

$$\begin{bmatrix} (n+1) & \sum_{i=0}^{n} \ln x_i \\ \sum_{i=0}^{n} \ln x_i & \sum_{i=0}^{n} (\ln x_i)^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} \ln y_i \\ \sum_{i=0}^{n} \ln x_i \ln y_i \end{bmatrix} \tag{8.19}$$

Desta forma, os parâmetros são $a = e^{a_0}$ e $b = a_1$.

8.5.3 Ajustamento por uma função hiperbólica

Neste caso, $y = \frac{1}{a_0 + a_1 x}$. A linearização desta função resulta em $\frac{1}{y} = a_0 + a_1 x$ e o sistema

$$\begin{bmatrix} (n+1) & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} \frac{1}{y_i} \\ \sum_{i=0}^{n} \frac{x_i}{y_i} \end{bmatrix}$$

8.5.4 Ajustamento por uma função do tipo $y = \frac{x}{a_0 + a_1 x}$

Neste caso, a linearização é $\frac{x}{y} = a_0 + a_1 x$ e o sistema obtido é dado por

$$\begin{bmatrix} (n+1) & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} \frac{x_i}{y_i} \\ \sum_{i=0}^{n} \frac{x_i^2}{y_i} \end{bmatrix} \tag{8.20}$$

8.5.5 Ajustamento por uma função do tipo $y = \frac{1}{a_0 + a_1 x + a_2 x^2}$

Neste caso, a linearização resulta $\frac{1}{y} = a_0 + a_1 x + a_2 x^2$ e o sistema é dado por

$$\begin{bmatrix} (n+1) & \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 \\ \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 & \sum_{i=0}^{n} x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} \frac{1}{y_i} \\ \sum_{i=0}^{n} \frac{x_i}{y_i} \\ \sum_{i=0}^{n} \frac{x_i^2}{y_i} \end{bmatrix} \tag{8.21}$$

8.5.6 Ajustamento por uma função do tipo $y = a\,e^{b x + c x^2}$

A linearização é empregada da seguinte forma: $Y = \ln y$, $X = x$, $a_0 = \ln a$, $a_1 = b$ e $a_2 = c$. Sendo assim, o sistema normal fica

$$\begin{bmatrix} (n+1) & \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 \\ \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 & \sum_{i=0}^{n} x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} \ln y_i \\ \sum_{i=0}^{n} x_i \ln y_i \\ \sum_{i=0}^{n} x_i^2 \ln y_i \end{bmatrix} \tag{8.22}$$

8.6 Escolha do melhor ajuste

Uma vez conhecidas as diferentes formas de regressão, podemos nos indagar: para um determinado conjunto de dados experimentais, qual é a *melhor* forma?

Essa pergunta pode ser respondida se considerarmos algumas medidas dos erros envolvidos nas regressões, essencialmente comparando o quão distante um valor experimental f_i está do valor y_i calculado através da equação para as diferentes regressões. Basicamente, podemos considerar três medidas diferentes:

Erro relativo: Seleciona-se a regressão que tiver o *menor* erro relativo máximo,

$$e = \max_{i=1}^{m} \frac{|f_i - y_i|}{|f_i|} \tag{8.23}$$

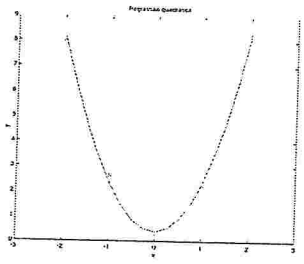

Figura 8.5: *Regressão quadrática.*

obtenha as quatro regressões – linear, quadrática (polinomial), potencial e exponencial – calculando as medidas para escolha da melhor regressão.
Solução: *Calculadas as regressões, com exceção da potencial (a qual não pode ser calculada sem translação devido à presença de um valor nulo nos dados experimentais), obtemos os seguintes valores para y_i (arredondados para a primeira casa decimal):*

x	1	2	3	4	5
linear	4,2	4,3	4,3	4,3	4,3
quadrática	8,2	2,3	0,4	2,3	8,2
exponencial	2,5	2,5	2,5	2,4	2,4

e os correspondentes valores das medidas

	e	t	D
linear	17,2885	0,0080	0,0032
quadrática	0,5724	0,9220	0,4275
exponencial	9,5135	0,0219	0,0086

Analisando a tabela, vemos que, se o critério escolhido fosse o erro relativo e, deveríamos escolher a regressão quadrática, com coeficientes

$$a_0 = 0,3675, \quad a_1 = 0,0171, \quad a_2 = 1,9536;$$

já para as outras duas medidas, a regressão escolhida seria a linear, com

$$a_0 = 4,2748, \quad a_1 = 0,0171$$

Analisando-se o gráfico na figura (8.5), observa-se que a regressão quadrática é melhor, evidentemente.

Os exemplos aqui apresentados mostram que o ajuste de dados experimentais é um processo numérico que deve ser usado tomando-se cuidado ao se selecionar uma dada regressão, se possível fazendo-se o gráfico dos dados experimentais e da curva de regressão.

8.7 Mínimos quadrados - domínio contínuo

Mesmo no caso em que a forma analítica da função f é conhecida, às vezes é de interesse aproximá-la no intervalo $I = [x_I, x_F]$ por uma função g da família

$$\sum_{k=0}^{m} a_k \, g_k(x) \tag{8.26}$$

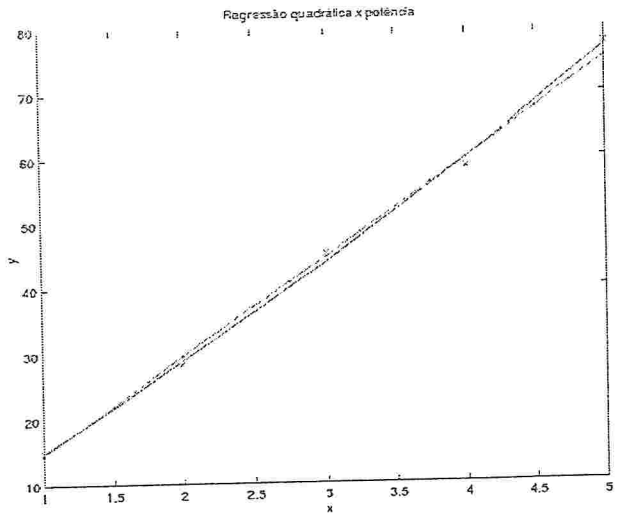

Figura 8.4: *Regressão quadrática versus potencial.*

Solução: *Calculadas as regressões, obtemos os seguintes valores para y_i (arredondados para a primeira casa decimal):*

x	1	2	3	4	5
linear	$-21,1$	$12,8$	$46,6$	$80,5$	$114,4$
quadrática	$7,5$	$-1,5$	$18,1$	$66,2$	$142,9$
potencial	$2,0$	$10,6$	$28,3$	$56,7$	$97,4$
exponencial	$2,7$	$7,4$	$20,1$	$54,6$	$148,4$

e os correspondentes valores das medidas

	e	t	D
linear	$1,5745$	$1,4519$	$0,5739$
quadrática	$1,7618$	$2,0648$	$0,6415$
potencial	$0,4331$	$1,4977$	$0,4973$
exponencial	$0,0000$	$2,1820$	$0,6476$

Analisando a tabela, vemos que, se o critério escolhido fosse o erro relativo e, deveríamos escolher a regressão exponencial, com coeficientes

$$a_0 = 1,0, \quad a_1 = 2,7183$$

o que é óbvio, pois os valores tabulados representam justamente $f_i = e^{x_i}$. Se, no entanto, utilizássemos como medida o desvio relativo em relação à média, t, escolheríamos a regressão linear, com

$$a_0 = -54,9388, \quad a_1 = 33,8599$$

que certamente não seria uma boa escolha; finalmente, escolhendo o coeficiente de variação da amostra, a regressão potencial seria escolhida, com

$$a_0 = 1,9785, \quad a_1 = 2,4216$$

Exemplo 8.8 *Dada a tabela abaixo*

x	-2	-1	0	1	2
f	$8,0064$	$2,6319$	$0,2337$	$2,1888$	$8,3132$

mais conveniente. Às vezes, por exemplo, tem-se uma função com descontinuidades, mas quer-se trabalhar com uma função contínua. A primeira vista, seria possível recair no caso discreto tabelando a função dada em alguns pontos; entretanto, isto pode causar perda de informação sobre o comportamento do erro.

Ao se considerar a soma dos quadrados dos resíduos em todos os pontos do intervalo $[x_I, x_F]$, tem-se, no limite, a integral do quadrado do resíduo em cada ponto do intervalo em que se quer aproximar a função dada. Geometricamente, isto representa a área entre as curvas $f(x)$ e $g(x)$.

Assim, é necessário determinar $a_0, a_1, \ldots, a_m$ que minimizam

$$\begin{aligned} M(a_0, a_1, \ldots, a_m) &= \int_{x_I}^{x_F} r^2(x)\,dx \\ &= \int_{x_I}^{x_F} (f(x) - g(x))^2\,dx \\ &= \int_{x_I}^{x_F} (f(x) - a_0\,g_0(x) - a_1\,g_1(x) - \ldots - a_m\,g_m(x))^2\,dx \end{aligned} \tag{8.27}$$

Como no caso discreto, o ponto de mínimo é atingido quando

$$\frac{\partial M}{\partial a_0} = \frac{\partial M}{\partial a_1} = \ldots = \frac{\partial M}{\partial a_m} = 0$$

ou seja,

$$\frac{\partial M}{\partial a_l} = -2 \int_{x_I}^{x_F} \left(f(x) - \sum_{k=0}^{m} a_k\,g_k(x) \right)^2 g_l(x)\,dx = 0 \qquad 0 \le l \le m. \tag{8.28}$$

Usando a definição de produto escalar de duas funções $w(x)$ e $q(x)$ no intervalo $[x_I, x_F]$ como

$$< w,\, q >= \int_{x_I}^{x_F} w(x)\,q(x)\,dx$$

tem-se que, no caso em que se quer aproximar $f(x)$, o sistema normal fica

$$\begin{bmatrix} < g_0, g_0 > & < g_0, g_1 > & \ldots & < g_0, g_m > \\ < g_1, g_0 > & < g_1, g_1 > & \ldots & < g_1, g_m > \\ \vdots & \vdots & \ddots & \vdots \\ < g_m, g_0 > & < g_m, g_1 > & \ldots & < g_m, g_m > \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} < g_0, f > \\ < g_1, f > \\ \vdots \\ < g_m, f > \end{bmatrix} \tag{8.29}$$

Exemplo 8.9 *Aproxime a função exponencial e^x no intervalo $[0,1]$ por uma reta utilizando o método dos mínimos quadrados.*

Neste caso, $g(x) = a_0 + a_1\,x$. Com a notação utilizada,

$$g_0(x) = 1 \qquad g_1(x) = x \qquad f(x) = e^x$$

e o produto escalar,

$$\int_0^1 f(x)\,g(x)\,dx.$$

Portanto, determinar a_0 e a_1 pelo método dos mínimos quadrados é calcular a solução do seguinte sistema normal:

$$\begin{bmatrix} < 1, 1 > & < 1, x > \\ < x, 1 > & < x, x > \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} < 1, e^x > \\ < x, e^x > \end{bmatrix}.$$

Família de funções	Condições
Reta - $y = a_0 + a_1\,x$	$f[x_{i+1}, x_i] \approx$ const.
Parábola - $y = a_0 + a_1\,x + a_2\,x^2$	$f[x_{i+2}, x_{i+1}, x_i] \approx$ const.
Função exponencial - $y = c\,e^{a\,x}$	$\dfrac{\Delta \ln y_i}{\Delta x_i} \approx$ const.
Função potência - $y = a\,x^b$	$\dfrac{\Delta \ln y_i}{\Delta \ln x_i} \approx$ const.
Função hiperbólica - $y = \dfrac{1}{a_0 + a_1\,x}$	$\dfrac{\Delta \frac{1}{y_i}}{\Delta x_i} \approx$ const.
Função do tipo $y = \dfrac{x}{a_0 + a_1\,x}$	$\dfrac{\Delta \frac{x_i}{y_i}}{\Delta x_i} \approx$ const.
Função do tipo $y = \dfrac{1}{a_0 + a_1\,x + a_2\,x^2}$	$\dfrac{\Delta \frac{1}{y_{i+1}} - \Delta \frac{1}{y_i}}{x_{i+2} - x_i} \approx$ const.
Função do tipo $y = a\,e^{b\,x + c\,x^2}$	$\dfrac{\Delta \ln y_{i+1} - \Delta \ln y_i}{x_{i+2} - x_i} \approx$ const.

Tabela 8.1: *Condições necessárias para se ajustar pontos.*

Sendo assim,

$$\begin{aligned} p_0(x) &= 1\,x^0 = 1 \\ p_1(x) &= 1\,x^1 + c_0\,x^0 = x + c_0 \\ p_2(x) &= 1\,x^2 + d_1\,x^1 + d_0\,x^0 = x^2 + d_1\,x + d_0 \end{aligned}$$

A constante c_0 *é determinada impondo-se* $< p_1, p_0 >= 0$, *ou seja,*

$$< p_1, p_0 >= \int_0^1 (x + c_0)\,1\,dx = \frac{1}{2} + c_0 = 0,$$

Portanto, como $c_0 = -\frac{1}{2}$, $p_1(x) = x - \frac{1}{2}$.

As outras duas constantes, d_1 *e* d_0 *são determinadas fazendo-se* $< p_2, p_0 >= 0$ *e* $< p_2, p_1 >= 0$, *de onde se obtém* $p_2(x) = x^2 - x + \frac{1}{6}$.

Agora, utiliza-se os polinômios ortogonais calculados para determinar o polinômio desejado, de grau 2:

$$g(x) = a_0\,p_0(x) + a_1\,p_1(x) + a_2\,p_2(x)$$

que aproxima $f(x) = e^x$ *no intervalo* $[0, 1]$.

De (8.32) *tem-se*

$$a_k = \frac{< p_k, f >}{< p_k, p_k >}$$

Logo,

$$a_0 = \frac{< 1, e^x >}{< 1, 1 >} = \frac{\int_0^1 e^x\,dx}{\int_0^1 dx} = e - 1$$

$$a_1 = \frac{< x - \frac{1}{2}, e^x >}{< x - \frac{1}{2}, x - \frac{1}{2} >} = \frac{\int_0^1 e^x \left(x - \frac{1}{2}\right) dx}{\int_0^1 \left(x - \frac{1}{2}\right)^2 dx} = 6\,(3 - e)$$

$$a_2 = \frac{< x^2 - x + \frac{1}{6}, e^x >}{< x^2 - x + \frac{1}{6}, x^2 - x + \frac{1}{6} >} = \frac{\int_0^1 e^x \left(x^2 - x + \frac{1}{6}\right) dx}{\int_0^1 \left(x^2 - x + \frac{1}{6}\right)^2 dx} = 30\,(7\,e - 19)$$

Portanto,

$$g(x) = (e - 1) + 6\,(3 - e)\left(x - \frac{1}{2}\right) + 30\,(7\,e - 19)\left(x^2 - x + \frac{1}{6}\right)$$

Um exemplo importante de uma família de polinômios ortogonais é a dos polinômios de Legendre, que obedecem à seguinte definição:

$$< p_n, p_m > = \int_{-1}^{1} p_n(x)\,p_m(x)\,dx \tag{8.36}$$

$$= \begin{cases} 0 & se \quad m \neq n \\ \dfrac{2}{2\,n + 1} & se \quad m = n \end{cases} \tag{8.37}$$

Como

$$\begin{aligned} < 1, 1 > &= \int_0^1 1\,dx = 1 \\ < 1, x > &= \int_0^1 x\,dx = \frac{1}{2} \\ < x, x > &= \int_0^1 x^2\,dx = \frac{1}{3} \\ < 1, e^x > &= \int_0^1 e^x\,dx = e - 1 \\ < x, e^x > &= \int_0^1 x\,e^x\,dx = 1 \end{aligned}$$

tem-se

$$\begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} e-1 \\ 1 \end{bmatrix}$$

A solução $a_0 = 4\,e - 10$ *e* $a_1 = 18 - 6\,e$ *determina a função aproximadora* $g(x) = 4\,e - 10 + (18 - 6\,e)\,x$.

8.7.1 Polinômios ortogonais

Quando se aproxima uma função f por uma função g da família

$$\sum_{k=0}^{m} a_k\, g_k(x) \tag{8.30}$$

pelo método dos mínimos quadrados, é necessário resolver um sistema linear de equações denominado sistema normal. Se um conjunto de funções $\{g_k\}$, $k = 0, 1, \ldots, m$ tais que

$$< g_k, g_l >= 0 \qquad \forall\, k \neq l, \quad 0 \leq k, l \leq m \tag{8.31}$$

o sistema normal se torna diagonal e os coeficientes a_k da função aproximadora são determinados por

$$a_k = \frac{< g_k, f >}{< g_k, g_k >}, \qquad 0 \leq k \leq m \tag{8.32}$$

As funções que satisfazem a relação (8.31) são denominadas *funções ortogonais*. Um polinômio de grau k pode ser escrito na forma $p_k(x) = c_k\,x^k + c_{k-1}\,x^{k-1} + \ldots + c_1\,x + c_0$. Os polinômios ortogonais $p_k(x)$, $k = 0, 1, \ldots$ obedecem às seguintes relações:

$$< p_k, p_l > \;=\; 0 \quad \text{para} \quad k \neq l \tag{8.33}$$

$$< p_k, p_k > \;>\; 0 \quad \text{para} \quad k = 0, 1, \ldots \tag{8.34}$$

(8.35)

Exemplo 8.10 *Construa os três primeiros polinômios ortogonais com relação ao produto escalar*

$$< f, g >= \int_0^1 f(x)\,g(x)\,dx$$

e aproxime $f(x) = e^x$ *no intervalo* $[0, 1]$ *por um polinômio de grau 2. Para este caso, imponha a condição de que o coeficiente do termo de mais alto grau de cada polinômio seja igual a 1.*

x	-1	0	1	2	3	4	5	6
y	10	9	7	5	4	3	0	-1

Exercício 8.2 *Encontre a função exponencial que melhor se ajusta aos pontos* $(0;1,5)$, $(1;2,5)$, $(2;3,5)$, $(3;5)$ *e* $(4;7,5)$.

Exercício 8.3 *Encontre a parábola que melhor se ajusta aos pontos* $(-3;3)$, $(0;1)$, $(2;1)$ *e* $(4;3)$.

Exercício 8.4 *Encontre a hipérbole que melhor se ajusta aos pontos* $(0;0,2)$, $(1;0,11)$, $(2;0,08)$, $(3;0,06)$ *e* $(4;0,05)$.

Exercício 8.5 *Considere a variação da viscosidade* η *em função da temperatura:*

T	7,5	10,9	14,0	15,0	16,0	18,0	21,0
η	1409	1276	1175	1148	1121	1069	990

Encontre a melhor função de ajustamento e determine a viscosidade para $T = 4\,°C$ *e* $T = 25\,°C$.

Exercício 8.6 *Admita que a venda de peixes de um determinado mercado seja conforme a tabela*

dia	1	5	10	15
número de peixes	70	30	55	25

Determine a função que melhor se ajusta aos dados.

Exercício 8.7 *Ajuste os dados da tabela utilizando*

1. *uma função exponencial;*
2. *uma função potência.*

x	1	2	3	4	5
y	0,6	1,9	4,3	7,6	12,6

Depois, utilize o critério dos mínimos quadrados para determinar qual das curvas, (1) ou (2), é melhor.

Exercício 8.8 *Considere a incidência de câncer, problemas cardíacos e complicações respiratórias em pacientes, conforme a idade, mostrados na tabela (por mil habitantes).*

idade	*incidência de câncer*	*problemas cardíacos*	*complicações respiratórias*
5	*0*	*0*	*1*
15	*0*	*1*	*3*
25	*1*	*5*	*5*
35	*3*	*12*	*7*
45	*6*	*30*	*10*
55	*12*	*79*	*12*
65	*30*	*140*	*14*

Identifique as funções que melhor se ajustam aos problemas indicados conforme a idade.

Exercício 8.9 *Encontre a constante de aceleração da gravidade* g *para o seguinte conjunto de dados:*

t	0,200	0,400	0,600	0,800	1,000
x	0,1960	0,7835	1,7630	3,1345	4,8975

Exercício 8.10 *Aproxime a função* $4\,x^3$ *por um polinômio de primeiro grau, uma reta, no intervalo* $[x_I, x_F] = [0,1]$.

Exercício 8.11 *Repita o exemplo 8.9 utilizando os polinômios ortogonais obtidos acima e verifique que o resultado obtido é o mesmo.*

Usando esta definição, pode-se construir os três primeiros polinômios de Legendre:

$$p_0(x) = 1, \qquad p_1(x) = x, \qquad p_2(x) = \frac{1}{2}(3x^2 - 1) \tag{8.38}$$

Além desta, ainda existem outras famílias de polinômios ortogonais, como os polinômios de Hermite, Chebyshev, etc. Estes polinômios, tabelados ou previamente calculados, podem ser empregados para ajustar uma função f por um polinômio g de grau menor ou igual a m em um intervalo $[a, b]$.

Supondo que se tenha a disposição uma tabela de polinômios ortogonais em um intervalo $[c, d]$, é preciso fazer uma mudança de variável $t(x) = \alpha x + \beta$ para transformar linearmente $f(t)$, definida no intervalo $[a, b]$, em $f(t(x)) = F(x)$ definida no intervalo $[c, d]$. Faz-se, então, o ajuste da função $F(x)$ por um polinômio $G(x)$ de grau menor ou igual a m usando os polinômios tabelados. Por transformação inversa de variável, $x(t) = \gamma t + \delta$, obtém-se a função aproximadora $g(t) = G(x(t))$.

Exemplo 8.11 *Aproxime a função* $f(t) = \operatorname{sen} t$ *no intervalo* $0 \leq t \leq \pi$ *por uma parábola, utilizando os polinômios de Legendre.*

Fazendo a mudança de variável que transforma linearmente o intervalo $[0, \pi]$ *em* $[-1, 1]$, *tem-se*

$$t(x) = \frac{\pi}{2}(x + 1)$$

Nestas condições,

$$f(t(x)) = \operatorname{sen} t(x) = \operatorname{sen}\left(\frac{\pi}{2}(x+1)\right) = F(x).$$

A parábola que se quer obter pelo método dos mínimos quadrados é

$$G(x) + a_0\, 1 + a_1 x + a_2 \frac{1}{2}(3x^2 - 1)$$

Como os polinômios de Legendre são ortogonais, emprega-se (8.32) *para determinar os coeficientes* a_0, a_1 *e* a_2 *da parábola:*

$$\begin{aligned} a_0 &= \frac{< F, p_0 >}{< p_0, p_0 >} = \frac{2}{\pi} \\ a_1 &= \frac{< F, p_1 >}{< p_1, p_1 >} = 0 \\ a_2 &= \frac{< F, p_2 >}{< p_2, p_2 >} = \frac{10}{\pi}\left[1 - \frac{12}{\pi^2}\right] \end{aligned}$$

Desta forma,

$$G(x) = \frac{2}{\pi} + \frac{10}{\pi}\left[1 - \frac{12}{\pi^2}\right]\frac{1}{2}(3x^2 - 1) \quad \textit{para} \quad x \in [-1, 1]$$

Voltando para o intervalo inicial $[0, \pi]$ *através da transformação inversa,* $x(t) = \frac{2}{\pi}t - 1$, *obtém-se*

$$g(t) = \frac{2}{\pi} + \frac{10}{\pi}\left[1 - \frac{12}{\pi^2}\right]\frac{1}{2}\left[3\left[\frac{2}{\pi}t - 1\right]^2 - 1\right]$$

que é a função aproximadora desejada.

8.8 Exercícios

Exercício 8.1 *Utilize o método dos mínimos quadrados para encontrar a reta que melhor se ajusta aos pontos da tabela.*

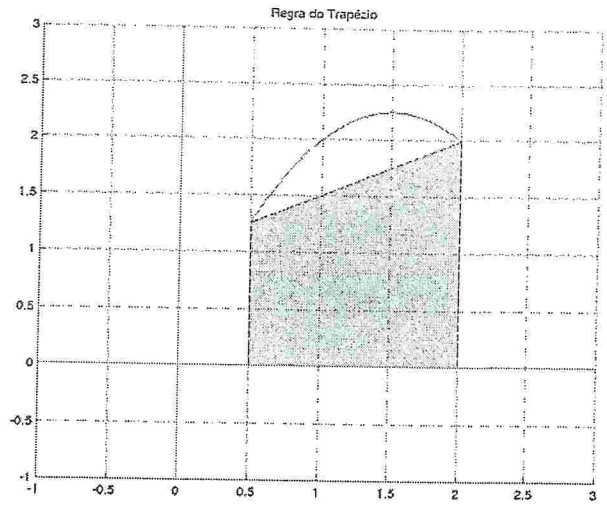

Figura 9.1: *A regra do trapézio.*

onde

$$A_i = \int_a^b l_i(x)\,dx$$

a qual é conhecida como a *forma de Newton-Cotes*, se os pontos x_i forem igualmente espaçados. A partir da equação (9.5), pode-se derivar várias regras de integração, dependendo do grau do polinômio de Lagrange.

9.2.1 Regra do Trapézio

Se tomarmos $n = 1$, e usarmos como nós os pontos extremos do intervalo, i.e. $x_0 = a$, $x_1 = b$, obtemos a chamada *regra do trapézio*. Nesse caso, os polinômios interpoladores são

$$l_0(x) = \frac{b-x}{b-a}, \quad l_1(x) = \frac{x-a}{b-a}$$

de onde

$$A_0 = \int_a^b l_0(x)\,dx = \frac{1}{2}(b-a) = \int_a^b l_1(x)\,dx = A_1$$

Assim, escrevendo a equação (9.5) para esse caso particular, temos

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\,(f(a)+f(b)) \tag{9.6}$$

a qual define a regra do trapézio. Essa fórmula é exata para qualquer polinômio de grau igual a 1, no máximo; o erro associado a essa aproximação é dado por

$$-\frac{1}{12}(b-a)^3 f''(\xi), \quad a < \xi < b \tag{9.7}$$

Ao usarmos a regra do trapézio, estamos substituindo a função f por uma reta, no intervalo $[a,b]$, conforme a figura 9.1. É claro que essa aproximação pode ser bastante crua, se $|\,b-a\,|$ é grande (o contrário também é verdade).

Capítulo 9

Integração Numérica

9.1 Introdução

A *integração numérica* é o processo computacional capaz de produzir um valor numérico para a integral de uma função sobre um determinado conjunto. Ela difere do processo de *antidiferenciação*, aprendido em Cálculo, na medida em que não se procura uma função F tal que $F' = f$; aqui, vamos procurar substituir f por uma outra função, g – tal que $f \approx g$ – mais amena à integração (por exemplo, g é um *polinômio*). Nesse caso, a solução numérica de

$$\int_a^b f(x)\,dx \tag{9.1}$$

será obtida calculando-se

$$\int_a^b g(x)\,dx, \quad g \approx f$$

Veremos, a seguir, o processo de integração numérica via *interpolação polinomial* e os diferentes métodos daí derivados.

9.2 Integração numérica via interpolação polinomial

Suponha a integral (9.1); podemos selecionar um conjunto de nós x_0, x_1, ..., x_n no intervalo $[a, b]$ e interpolar a função $f(x)$ através dos polinômios de Lagrange, os quais são expressos como

$$p(x) = \sum_{i=0}^{n} f(x_i) l_i(x) \tag{9.2}$$

onde

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, 1, \ldots, n \tag{9.3}$$

Agora, substituímos $f(x)$ por $p(x)$, de tal forma que

$$\int_a^b f(x)\,dx \approx \int_a^b p(x)\,dx = \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)\,dx \tag{9.4}$$

a qual pode ser usada para calcular a integral de *qualquer* função. A equação acima pode ser reescrita na forma

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} A_i f(x_i) \tag{9.5}$$

1. *Usando a fórmula (9.8), para* $n = 2$ *e usando* $x_0 = 1$, $x_1 = 1,1$ *e* $x_2 = 2$, *temos*

$$A = \frac{1}{2}[(1,1-1)(4+4,51) + (2-1,1)(4,51+10)] = 6,9550$$

e o erro, comparado com o valor da integral definida ($= 6,8333$*), é de* $-0,1217$.

2. *Usando a fórmula (9.9), para* $n = 2$ *e usando* $x_0 = 1$, $x_1 = 1,5$ *e* $x_2 = 2$, *temos*

$$A = \frac{1}{2}[(1,5-1)(4+6,75) + (2-1,5)(6,75+10)] = 6,8750$$

e o erro, comparado com o valor da integral definida ($= 6,8333$*), é de* $-0,0417$.

Note que, em ambos os casos, a aproximação com a regra composta é melhor do que usando a regra simples do trapézio.

Exemplo 9.3 *Considere a tabela abaixo, que fornece a velocidade (km/h) de um certo objeto em função do tempo e determine qual é a distância percorrida pelo objeto ao final de* $2\,h$.

t	0,00	0,25	0,50	0,75	1,00	1,25	1,50	1,75	2,00
$v(t)$	6,0	7,5	8,0	9,0	8,5	10,5	9,5	7,0	6,0

Como a distância percorrida (d) é calculada como

$$d = \int_0^2 v(t)\,dt,$$

pode-se empregar a regra dos trapézios com $n = 8$, $h = 0,25$, *de forma que*

$$A = \frac{0,25}{2}\left[6 + 2\,(7,5+8,0+9,0+8,5+10,5+9,5+7,0)) + 6\right].$$

Portanto, uma aproximação para a distância total percorrida no intervalo de tempo $[0,2]$ *é*

$$d \approx A = 16,5km.$$

Exemplo 9.4 *Considere as integrais definidas*

$$\int_1^3 \frac{x}{1+x^2}\,dx \qquad e \qquad \int_1^3 \frac{dx}{7-2\,x}$$

As tabelas 9.1 *e* 9.2 *mostram as aproximações obtidas usando a regra dos trapézios com* $n = 1,2,4,8,16,32$ *subintervalos e o erro na aproximação. Note que, à medida que n cresce, h é sucessivamente dividido por* 2 *e cada erro é aproximadamente* $\frac{1}{4}$ *do erro anterior.*

n	h	A	$E(f,h)$
1	2,0	0,800000	0,004719
2	1,0	0,800000	0,004719
4	0,5	0,803183	0,001536
8	0,25	0,804311	0,000408
16	0,125	0,804615	0,000104
32	0,0625	0,804693	0,000026

Tabela 9.1: Aproximação para $\int_1^3 \frac{x}{1+x^2}\,dx$.

Exemplo 9.1 *Calcule a integral*

$$A = \int_1^2 x^2 + 3x\,dx$$

usando a regra do trapézio.
Solução: *Usando a fórmula (9.6), temos*

$$A = \frac{2-1}{2}(4+10) = \frac{14}{2} = 7$$

Como a antiderivada $F(x) = \frac{x^3}{3} + \frac{3x^2}{2}$ *é conhecida, podemos avaliar o erro. Calculando a integral definida, temos*

$$\int_1^2 x^2 + 3x\,dx = \frac{x^3}{3} + \frac{3x^2}{2}\Big|_1^2 = 6,8333$$

de onde podemos calcular o erro como sendo igual a $6,8333 - 7 = -0,1667$. *Usando a fórmula (9.7), com* $f'' = 2$, *obtemos o valor*

$$-\frac{1}{12}(2-1)^3 2 = -\frac{1}{6} = -0,1667$$

o qual é igual ao calculado anteriormente.

Podemos, evidentemente, obter uma melhor aproximação se *subdividirmos* o intervalo $[a,b]$, calculando nós $x_0, x_1, \ldots, x_n$ satisfazendo

$$a = x_0 < x_1 < \ldots < x_n = b$$

e aplicando a regra do trapézio a cada subintervalo (não necessariamente de mesmo tamanho). Essa estratégia nos leva à *regra composta do trapézio,*

$$\begin{aligned} \int_a^b f(x)\,dx &= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,dx \\ &\approx \frac{1}{2}\sum_{i=1}^{n} (x_i - x_{i-1})\,(f(x_{i-1}) + f(x_i)) \end{aligned} \tag{9.8}$$

A regra composta do trapézio nos leva à aproximação da função $f(x)$ por um conjunto de retas unindo cada um dos nós x_i, dois a dois, conforme a figura 9.2-a.

Se o espaçamento entre os nós é igual, i.e. $x_i = a + ih$, $h = \frac{b-a}{n}$, então obtemos a *regra composta uniforme do trapézio,*

$$T(f,h) = \int_a^b f(x)\,dx \approx \frac{h}{2}\left(f(a) + \left(2\sum_{i=1}^{n-1} f(a+ih)\right) + f(b)\right) \tag{9.9}$$

conforme a figura 9.2-b. O *erro de truncamento* $E(f,h)$ associado a essa aproximação é estimado por

$$E(f,h) \leq \frac{h^2}{12}(b-a) \max_{x\in[a,b]} |\, f''(x)\,|. \tag{9.10}$$

Exemplo 9.2 *Calcule a integral*

$$A = \int_1^2 x^2 + 3x\,dx$$

usando as regras composta e composta uniforme do trapézio.
Solução:

9.2.3 Regra de Simpson

A *regra de Simpson* é obtida a partir do método dos coeficientes a determinar, generalizada para um intervalo de integração $[a,b]$ qualquer. Ela é obtida a partir da integral de um polinômio interpolador de segundo grau $p_2(x)$ que passa por três pontos igualmente espaçados, $(a,f(a))$, $(m,f(m))$, e $(b,f(b))$, onde $m=(a+b)/2$. Assim, tomando $h=\frac{b-a}{2}$, tem-se

$$\int_a^b p_2(x)\,dx = \int_a^b \left[f(a)+(x-a)\frac{\Delta f(a)}{h}+(x-a)(x-m)\frac{\Delta^2 f(a)}{2\,h^2}\right]dx \tag{9.11}$$

Para facilitar o cálculo, faz-se a mudança de variável $x(\alpha)=a+\alpha h$. Assim, enquanto x percorre o intervalo $[a,b]$, α percorre o intervalo $[0,2]$ e $dx=h\,d\alpha$. Desta maneira,

$$\begin{aligned}\int_a^b p_2(x)\,dx &= \int_0^2 \left[f(a)+\alpha\,\Delta f(a)+\alpha\,(\alpha-1)\frac{\Delta^2 f(a)}{2}\right]h\,d\,\alpha \\ &= \frac{h}{3}\,[f(a)+4\,f(m)+f(b)]\end{aligned} \tag{9.12}$$

de onde a fórmula de Simpson pode ser escrita como

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\,[f(a)+4\,f(m)+f(b)] = \frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right) \tag{9.13}$$

a qual é exata para polinômios de grau $n\leq 2$ (conforme visto na seção anterior) e, inesperadamente, também para $n\leq 3$. O erro associado à regra de Simpson é

$$-\frac{1}{90}(b-a)^5 f^{(4)}(\xi), \quad a<\xi<b \tag{9.14}$$

Usando a mesma estratégia da regra composta uniforme do trapézio, podemos obter a *regra composta uniforme de Simpson*, para um número n *par*[1] de subintervalos. Nesse caso, temos

$$\begin{aligned}\int_a^b f(x)\,dx &= \int_{x_0}^{x_2} f(x)\,dx+\int_{x_2}^{x_4} f(x)\,dx+\ldots+\int_{x_{n-2}}^{x_n} f(x)\,dx \\ &= \sum_{i=1}^{\frac{n}{2}}\int_{x_{2i-2}}^{x_{2i}} f(x)\,dx\end{aligned}$$

de onde, aplicando a regra de Simpson a cada um dos subintervalos, obtemos

$$S(f,h)=\int_a^b f(x)\,dx \approx \frac{h}{3}\left(f(x_0)+2\sum_{i=2}^{\frac{n}{2}} f(x_{2i-2})+4\sum_{i=1}^{\frac{n}{2}} f(x_{2i-1})+f(x_n)\right) \tag{9.15}$$

O erro associado é

$$-\frac{1}{180}(b-a)h^4 f^{(4)}(\xi), \quad a<\xi<b \tag{9.16}$$

Exemplo 9.5 *Calcule a integral*

$$A=\int_1^2 x^2+3x\,dx$$

usando a regra de Simpson.
Solução: *Usando a fórmula (9.13), temos*

$$A=\frac{2-1}{6}[4+4\cdot 6,75+10]=6,8333$$

e o erro é nulo, *comparado com o valor da integral definida (= 6,8333). Note que, para a função em questão, $f^{(4)}=0$ e, portanto, a aproximação da integral pela regra de Simpson deve ser exata.*

[1] É necessária essa restrição devido à forma como a regra de Simpson foi definida.

n	h	A	$E(f,h)$
1	2,0	1,200000	−0,395281
2	1,0	0,933333	−0,128614
4	0,5	0,841667	−0,036948
8	0,25	0,814484	−0,009765
16	0,125	0,807203	−0,002484
32	0,0625	0,805343	−0,000624

Tabela 9.2: Aproximação para $\int_1^3 \frac{dx}{7-2x}$.

9.2.2 Método dos Coeficientes a Determinar

A equação (9.5) (fórmula de Newton-Cotes) é um caso particular do *método dos coeficientes a determinar*.

Suponha, por exemplo, que $n = 2$ e $[a,b] = [0,1]$. Nesse caso, os polinômios de Lagrange, escritos para os nós 0, $\frac{1}{2}$ e 1, são

$$l_0(x) = 2(x - \frac{1}{2})(x-1), \quad l_1(x) = -4x(x-1), \quad l_2(x) = 2x(x - \frac{1}{2}) \quad .$$

de onde podemos escrever

$$\begin{aligned} A_0 &= \int_0^1 l_0\,dx = \frac{1}{6} \\ A_1 &= \int_0^1 l_1\,dx = \frac{2}{3} \\ A_2 &= \int_0^1 l_2\,dx = \frac{1}{6} \end{aligned}$$

Os mesmos coeficientes A_i podem ser obtidos usando o método aqui descrito. Suponha que

$$\int_0^1 f(x)\,dx \approx A_0 f(0) + A_1 f\left(\frac{1}{2}\right) + A_2 f(1)$$

a qual deve ser exata para qualquer polinômio de grau igual ou inferior a 2. Para determinar os coeficientes, usamos as funções base 1, x e x^2 – i.e., $p(x) = c_0 + c_1 x + c_2 x^2$ – e escrevemos

$$\begin{aligned} \int_0^1 dx &= 1 = A_0 + A_1 + A_2 \\ \int_0^1 x\,dx &= \frac{1}{2} = \frac{1}{2}A_1 + A_2 \\ \int_0^1 x^2\,dx &= \frac{1}{3} = \frac{1}{4}A_1 + A_2 \end{aligned}$$

o que nos leva ao sistema de equações lineares

$$\left\{ \begin{array}{rcl} A_0 + A_1 + A_2 &=& 1 \\ \frac{1}{2}A_1 + A_2 &=& \frac{1}{2} \\ \frac{1}{4}A_1 + A_2 &=& \frac{1}{3} \end{array} \right.$$

o qual tem a seguinte solução: $A_0 = \frac{1}{6}$, $A_1 = \frac{2}{3}$, $A_2 = \frac{1}{6}$.

1. *Cálculo de $S(1)$: Para calcular a primeira aproximação, $S(1)$, é preciso conhecer $T(0)$ e $T(1)$:*

   (a) *Cálculo de $T(0)$: se $J = 0$, conseqüentemente $h = b - a = 4$. Logo,*

   $$T(0) = 4\,\frac{\frac{1}{1} + \frac{1}{5}}{2} = 2,4$$

   (b) *Cálculo de $T(1)$: se $J = 1$, conseqüentemente $n = 1$ e $h = \frac{b-a}{2^1} = 2$. Logo, com $x_1 = a + h = 3$,*

   $$T(1) = \frac{T(0)}{2} + h\,f(x_1) = \frac{2,4}{2} + 2\,\frac{1}{3} = 1,866666$$

   Assim,

   $$S(1) = \frac{4\,T(1) - T(0)}{3} = 1,688888$$

2. *Cálculo de $S(2)$: como $T(1)$ já é conhecido, calcula-se apenas $T(2)$ com $n = 2$, $h = \frac{b-a}{2^2} = 1$, $x_1 = a + h = 2$ e $x_3 = a + 3\,h = 4$:*

   $$\begin{aligned} T(2) &= \frac{T(1)}{2} + \sum_{k=1}^{2}[f(x_1) + f(x_3)] \\ &= \frac{1,866666}{2} + \left[\frac{1}{2} + \frac{1}{4}\right] \\ &= 1,683333 \end{aligned}$$

   de forma que

   $$S(2) = \frac{4\,T(2) - T(1)}{3} = 1,622222$$

3. *Cálculo de $S(3)$: como $T(2)$ já é conhecido, calcula-se $T(3)$ com $n = 4$, $h = \frac{b-a}{2^3} = 0,5$, $x_1 = a + h = 1,5$, $x_3 = a + 3\,h = 2,5$, $x_5 = a + 5\,h = 3,5$ e $x_7 = a + 7\,h = 4,5$:*

   $$\begin{aligned} T(3) &= \frac{T(2)}{2} + \sum_{k=1}^{4}[f(x_1) + f(x_3) + f(x_5) + f(x_7)] \\ &= \frac{1,683333}{2} + 0,5\left[\frac{1}{1,5} + \frac{1}{2,5} + \frac{1}{3,5} + \frac{1}{4,5}\right] \\ &= 1,628968 \end{aligned}$$

   ou seja,

   $$S(3) = \frac{4\,T(3) - T(2)}{3} = 1,610846$$

9.2.5 Mudança do intervalo de integração

Algumas regras de integração são definidas em termos de um intervalo de integração fixo – por exemplo, $[-1, 1]$. Caso se deseje utilizar uma dessas regras para se resolver a integral (9.1), pode-se proceder a uma mudança *linear* de variáveis.

Suponha uma regra de integração numérica dada por

$$\int_c^d f(t)\,dt \approx \sum_{i=0}^{n} A_i f(t_i) \tag{9.20}$$

Exemplo 9.6 *Use a fórmula de Simpson para encontrar a área sob a curva* $y = f(x)$ *que passa sob os três pontos* $(0,2)$, $(1,3)$ *e* $(2,2)$.

Como $n = 1$ *e* $h = 1$, *calcula-se*

$$\text{área} \approx S(f,h) = \frac{h}{3}\,[f(0) + 4f(1) + f(2)\,] = \frac{1}{3}\,[2 + 12 + 2] = \frac{16}{3}.$$

Exemplo 9.7 *O volume de um sólido de revolução é dado por*

$$\text{volume} = \pi \int_a^b [R(x)]^2 dx,$$

onde o sólido é obtido pela rotação da região sob a curva $y = R(x)$, $a \le x \le b$, *em torno do eixo* x. *Use a fórmula de Simpson para aproximar o volume do sólido de revolução, onde o raio* $R(x)$ *da posição ao longo do eixo* x *é dado na tabela*

x	0	1	2	3	4	5	6
$R(x)$	6,2	5,8	4,0	4,6	5,0	7,6	8,2

Usando a regra de Simpson com $n = 3$ *e* $h = 1$, *o valor aproximado da integral é calculado por*

$$\begin{aligned}
\text{volume} &\approx \frac{\pi}{3}\,[f(x_0)^2 + 4\,(f(x_1)^2 + f(x_3)^2 + f(x_5)^2) + 2\,(f(x_2)^2 + f(x_4)^2) + f(x_6)^2] \\
&\approx \frac{\pi}{3}\,[(6,2)^2 + 4\,((5,8)^2 + (4,6)^2 + (7,6)^2) + 2\,((4,0)^2 + (5,0)^2) + (8,2)^2] \\
&\approx \frac{\pi}{3}\,[38,44 + 4\,(33,64 + 21,16 + 57,76) + 2\,(16,00 + 25,00) + 67,24] \\
&\approx 668,03
\end{aligned}$$

9.2.4 Regra de Simpson com exatidão crescente

Esta regra calcula uma aproximação por Simpson com uma combinação linear de fórmulas dos trapézios, $\{T(J)\}$. Para $J \ge 1$, divide-se o intervalo $[a,b]$ em $2n = 2^J$ subintervalos de igual espaçamento $h = \frac{b-a}{2^J}$ e usa-se os pontos $a = x_0 < x_1 < \ldots < x_{2n} = b$, $x_k = a + hk$ para $k = 0, 1, \ldots, 2n$. A regra dos trapézios $T(f,h)$ e $T(f,2h)$ para espaçamentos h e $2h$, respectivamente, obedece a relação

$$T(f,h) = \frac{T(f,2h)}{2} + h \sum_{k=1}^{n} f(x_{2k-1}). \tag{9.17}$$

Definindo $T(0) = \frac{h}{2}(f(a) + f(b))$, então para qualquer inteiro positivo J define-se $T(J) = T(f,h)$ e $T(J-1) = T(f,2h)$, o que permite escrever a fórmula acima como

$$T(J) = \frac{T(J-1)}{2} + h \sum_{k=1}^{n} f(x_{2k-1}) \qquad \text{para} \qquad J = 1, 2, \ldots \tag{9.18}$$

Assim, a regra de Simpson $S(J) = S(f,h)$ para 2^J subintervalos é obtida de $T(J)$ e de $T(J-1)$ pela fórmula

$$S(J) = \frac{4\,T(J) - T(J-1)}{3} \qquad \text{para} \qquad J \ge 1 \tag{9.19}$$

Exemplo 9.8 *Use a regra de Simpson com exatidão crescente para calcular aproximações* $S(1)$, $S(2)$ *e* $S(3)$ *para*

$$\int_1^5 \frac{dx}{x}$$

Solução: *Neste caso,* $a = 1$, $b = 5$ *e* $f(x) = \frac{1}{x}$.

a qual é exata para polinômios de grau igual ou inferior a m. Considere, agora, que o intervalo de integração desejado é $[a, b]$; para usarmos a fórmula (9.20), devemos definir uma função $\lambda(t)$ que associe c a a e d a b. Essa função pode ser dada por

$$\lambda(t) = \frac{b-a}{d-c}t + \frac{ad-bc}{d-c}, \quad c \leq t \leq d \tag{9.21}$$

Escrevendo, agora, $x = \lambda(t)$, temos $dx = \lambda'(t)\,dt = (b-a)(d-c)^{-1}\,dt$, de onde escrevemos a integral (9.1) como

$$\begin{aligned}\int_a^b f(x)\,dx &= \frac{b-a}{d-c}\int_{\lambda^{-1}(a)=c}^{\lambda^{-1}(b)=d} f(\lambda(t))\,dt \\ &\approx \frac{b-a}{d-c}\sum_{i=0}^{n} A_i f(\lambda(t_i))\end{aligned}$$

de onde

$$\int_a^b f(x)\,dx \approx \frac{b-a}{d-c}\sum_{i=0}^{n} A_i f\left(\frac{b-a}{d-c}t_i + \frac{ad-bc}{d-c}\right) \tag{9.22}$$

A função de transformação $\lambda(t)$ deve ser linear de forma que $f(\lambda(t))$ seja polinomial e de mesmo grau que f.

9.2.6 Quadratura Gaussiana

As regras de integração vistas nas seções anteriores são todas baseadas na determinação de coeficientes A_i tal que a aproximação da função integranda f é exata para polinômios de grau igual ou inferior a n.

No entanto, é possível escolher outros nós que levem a uma redução no volume de cálculo necessário. Por exemplo, se $A_i = c, \forall 0 \leq i \leq n$, então a forma de Newton-Cotes (9.5) pode ser escrita como

$$\int_a^b f(x)\,dx \approx c\sum_{i=0}^{n} f(xi) \tag{9.23}$$

o que elimina n multiplicações no processo de integração numérica.

As formas de *quadratura de Chebyshev* são um exemplo da equação (9.23); elas existem apenas para $n = 0, 1, 2, 3, 4, 5, 6$ e 8. Outras formas de quadratura existem, como, por exemplo, as de *Hermite* e as de *Gauss*.

A regra de integração de Gauss é expressa para o caso geral como

$$\int_a^b f(x)w(x)\,dx \approx \sum_{i=0}^{n} A_i f(x_i) \tag{9.24}$$

onde w é uma função positiva de ponderação. Assumindo que (9.24) é exata para qualquer função polinomial de grau menor ou igual a n, isso nos leva a determinar os coeficientes A_i como

$$A_i = \int_a^b w(x)\prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}\,dx$$

Carl Friedrich Gauss (1777-1855) mostrou que é possível determinar-se esses coeficientes de tal forma que a aproximação para f seja exata para polinômios de grau igual ou inferior a $2n+1$, mas com apenas n avaliações.

As fórmulas de Gauss para a integração de f são exatas para polinômios de grau menor ou igual a $2n+1$, de forma que a determinação dos pontos $x_0, x_1, \ldots, x_n$ em que é necessário

conhecer o valor de $f(x)$ será função do grau do polinômio interpolador e da fórmula específica a ser considerada. Estas formulas são do tipo

$$\int_a^b f(x)\,dx = w_0\,f(x_0) + w_1\,f(x_1) + \ldots + w_n\,f(x_n) \tag{9.25}$$

Para construir a fórmula da quadratura gaussiana para $n = 1$ é necessário determinar w_0, w_1, x_0 e x_1 tais que

$$\int_a^b f(x)\,dx = w_0\,f(x_0) + w_1\,f(x_1) \tag{9.26}$$

seja exata para polinômios de grau menor ou igual a 3.

Para simplificar os cálculos, determina-se esta fórmula considerando $[a,b] = [-1,1]$. No caso de um intervalo $[a,b]$ genérico efetua-se a mudança de variáveis: para $t \in [-1,1]$ corresponde $x \in [a,b]$ onde

$$x = \frac{1}{2}\,[a + b + t\,(b-a)] \qquad \text{e} \qquad dx = \frac{b-a}{2}\,dt$$

de forma que

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^1 F(t)\,dt \tag{9.27}$$

onde $F(t) = f(x(t))$.

Dizer que a fórmula é exata para polinômios de grau menor ou igual a 3 equivale a dizer que a fórmula é exata para

$$g(t) \equiv 1, \qquad g(t) \equiv t, \qquad g(t) \equiv t^2 \text{e} \qquad g(t) \equiv t^3$$

ou seja

$$\begin{aligned}
\int_{-1}^1 1\,dt &= w_0\,g(t_0) + w_1\,g(t_1) = w_0 + w_1 = 2 \\
\int_{-1}^1 t\,dt &= w_0\,g(t_0) + w_1\,g(t_1) = w_0\,t_0 + w_1\,t_1 = 0 \\
\int_{-1}^1 t^2\,dt &= w_0\,g(t_0) + w_1\,g(t_1) = w_0\,t_0^2 + w_1\,t_1^2 = 2/3 \\
\int_{-1}^1 t^3\,dt &= w_0\,g(t_0) + w_1\,g(t_1) = w_0\,t_0^3 + w_1\,t_1^3 = 0
\end{aligned}$$

Desta forma, obtém-se o seguinte sistema não linear:

$$\left\{\begin{array}{lcl}
w_0 + w_1 & = & 2 \\
w_0\,t_0 + w_1\,t_1 & = & 0 \\
w_0\,t_0^2 + w_1\,t_1^2 & = & 2/3 \\
w_0\,t_0^3 + w_1\,t_1^3 & = & 0
\end{array}\right. \tag{9.28}$$

cuja solução fornece

$$t_0 = -\frac{\sqrt{3}}{3} \qquad t_1 = \frac{\sqrt{3}}{3} \qquad w_0 = w_1 = 1,$$

Assim, a fórmula gaussiana para $n = 1$ é

$$\int_{-1}^1 F(t)\,dt = F\left(-\frac{\sqrt{3}}{3}\right) + F\left(\frac{\sqrt{3}}{3}\right) \tag{9.29}$$

O mesmo procedimento pode ser usado para determinar a fórmula geral (9.25). Supondo que $F(t)$ represente os polinômios especiais t^k para $k = 0, 1, \ldots, 2n+1$, observa-se que

$$\int_{-1}^{1} t^k \, dt = \begin{cases} 0 & \text{se } k \text{ é ímpar} \\ \dfrac{2}{k+1} & \text{se } k \text{ é par} \end{cases} \tag{9.30}$$

e a solução do sistema não linear que se origina destas equações é bastante complicada. Usando então a teoria dos polinômios ortogonais, pode ser visto que os t_k são as raízes de polinômios de Legendre[2] e os coeficientes w_k devem ser obtidos pela solução do sistema de equações. Alguns dos valores de t_k e w_k são mostrados na tabela 9.3; para quadraturas de maior ordem, pode-se recorrer aos valores tabelados em vários livros de referência.

n	t_k	w_k	k
1	−0,57735027	1,00000000	0
	0,57735027	1,00000000	1
2	−0,77459667	0,55555555	0
	0,00000000	0,88888888	1
	0,77459667	0,55555555	2
3	−0,86113631	0,34785485	0
	−0,33998104	0,65214515	1
	0,86113631	0,34785485	2
	0,33998104	0,65214515	3
4	−0,90617985	0,23692689	0
	−0,53846931	0,47862867	1
	0,00000000	0,56888889	2
	0,90617985	0,23692689	3
	0,53846931	0,47862867	4

Tabela 9.3: *Pesos e nós da quadratura Gaussiana, para* $n = 1, 2, 3, 4$.

O erro associado à quadratura Gaussiana é dado pela fórmula

$$\frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b q^2(x) w(x) \, dx, \quad q(x) = \prod_{i=0}^{n-1} (x - x_i), a < \xi < b \tag{9.31}$$

O algoritmo 9.2.1 faz uso da técnica de troca de intervalos e da simetria entre os nós e coeficientes, a fim de se calcular a integral (9.1) através da quadratura Gaussiana para $n = 4$. Na prática, a execução do algoritmo que calcule a integral (9.1) por quadratura Gaussiana sempre incorrerá em erros de ponto-flutuante, principalmente se os valores dos nós e coeficientes não forem utilizados com uma precisão adequada, como pode ser visto no exemplo a seguir.

[2]Os polinômios de Legendre são definidos pela seguinte fórmula de recorrência:

$$\begin{aligned} p_0(x) &= 1 \\ p_1(x) &= x \\ p_{m+1}(x) &= \frac{1}{m+1} \{(2m+1)\, x\, p_m(x) - m\, p_{m-1}(x)\}, \quad m = 1, 2, \ldots \end{aligned}$$

Suas raízes são todas reais e distintas e situam-se no intervalo $[-1, 1]$. Estas raízes estão simetricamente situadas com respeito à origem e se m é ímpar, uma raiz de $p_m(x)$ é sempre $x = 0$.

Algoritmo 9.2.1 *Quadratura Gaussiana de 4 pontos*

proc quadratura_gaussiana_4(input: a, b, f; output: S)
$x_0 \leftarrow 0$
$x_1 \leftarrow 0,5384\,6931\,0105\,683$
$x_2 \leftarrow 0,9061\,7984\,5938\,664$
$w_0 \leftarrow 0,5688\,8888\,8888\,889$
$w_1 \leftarrow 0,4786\,2867\,0499\,366$
$w_2 \leftarrow 0,2369\,2688\,5056\,189$
$u \leftarrow ((b-a)x_0 + a + b)/2$
$S \leftarrow w_0 f(u)$
for $i = 1, 2$ *do*
$u \leftarrow ((b-a)x_i + a + b)/2$
$v \leftarrow (-(b-a)x_i + a + b)/2$
$S \leftarrow S + w_i(f(u) + f(v))$
endfor
$S \leftarrow (b-a)S/2$
endproc

Exemplo 9.9 *Calcule a integral*

$$A = \int_1^2 x^2 + 3x\,dx$$

usando a quadratura de Gauss, com $n = 4$.
Solução: *Usando o algoritmo 9.2.1, temos*

$$A = 6,833333335$$

e o erro é igual a -2×10^{-9}, *comparado com o valor da integral definida* $(= 6,8333)$.

Exemplo 9.10 *Integre* $f(t) = t^4+1$ *no intervalo* $(-1, 1)$ *usando quadratura gaussiana para* $n = 2$.

$$I = \int_{-1}^{1} (t^4 + 1)\,dt = w_0\,f(t_0) + w_1\,f(t_1) + w_2\,f(t_2)$$

Da tabela 9.3, sabe-se que

$$\begin{aligned} t_0 &= -0,77459667 & w_0 &= 0,55555555 \\ t_1 &= 0,00000000 & w_1 &= 0,88888888 \\ t_2 &= 0,77459667 & w_2 &= 0,55555555 \end{aligned}$$

Logo,

$$\begin{aligned} I &= 0,55555556\,\left((-0,77459667)^4 + 1\right) \\ &\quad + 0,88888889\,\left((0,00000000)^4 + 1\right) \\ &\quad + 0,55555556\,\left((0,77459667)^4 + 1\right) \\ &= 2,4 \end{aligned}$$

Sugestão: *Calcule esta integral com o método de Simpson e compare os resultados.*

Exemplo 9.11 *Use quadratura gaussiana com três pontos para aproximar a integral*

$$\int_1^5 \frac{dx}{x}$$

Como o intervalo é $I = [1,5]$, é preciso fazer mudança de variável. Por isto, calcula-se a integral desejada como

$$\int_1^5 \frac{dx}{x} = \frac{b-a}{2} \int_{-1}^1 F(t)\,dt$$

com a mudança de variável

$$x = t\left(\frac{b-a}{2}\right) + \frac{a+b}{2} = t\left(\frac{5-1}{2}\right) + \frac{5+1}{2} = 2t+3$$

$$\begin{aligned}\int_1^5 \frac{dx}{x} &\approx \frac{5-1}{2}\left[w_0 F(t_0) + w_1 F(t_1) + w_2 F(t_2)\right] \\ &\approx \frac{5-1}{2}\left[0,55555556\,\frac{1}{2t_0+3} + 0,88888889\,\frac{1}{2t_1+3} + 0,55555556\,\frac{1}{2t_2+3}\right] \\ &\approx 1,602694\end{aligned}$$

onde $t_0 = -0,77459667$, $t_1 = 0,00000000$ e $t_2 = 0,77459667$.

9.3 Integração de funções mal comportadas

Funções mal comportadas (ou mal condicionadas) são aquelas que possuem algum tipo de característica especial e que, portanto, requerem cuidados especiais quando se quer integrá-las.

Exemplo 9.12 *Calcule a integral de*

$$\int_0^1 \frac{e^x}{\sqrt{x}}\,dx.$$

Solução: *Como esta função tem uma singularidade, é preciso fazer uma mudança de variável que a elimine. Neste caso, pode-se fazer*

$$x = u^2 \qquad e \qquad dx = 2u\,du$$

de forma que

$$\begin{aligned}\int_0^1 \frac{e^x}{\sqrt{x}}\,dx &= 2\int_0^1 \frac{e^{u^2}}{u} u\,du \\ &= 2\int_0^1 e^{u^2}\,du\end{aligned}$$

Como o integrando agora é uma função bem comportada, pode-se escolher um dos métodos estudados para calcular esta última integral.

Exemplo 9.13 *Calcule*

$$\int_0^1 \sqrt{\operatorname{sen} x}\,dx$$

Solução: *Como o integrando possui uma tangente vertical, a velocidade de integração fica muito lenta. Se o método escolhido fosse trapézios, por exemplo, seriam necessárias mais de 500 subdivisões do intervalo de integração $[0,1]$ para que se obtivesse quatro casas decimais repetidas.*

Neste caso, também é possível fazer a mudança de variável,

$$\operatorname{sen} x = u^2 \qquad e \qquad dx = \frac{2u\,du}{\sqrt{1-u^4}},$$

de maneira que

$$\int_0^1 \sqrt{\operatorname{sen} x}\, dx = 2 \int_0^{\sqrt{\operatorname{sen} 1}} \frac{u^2}{\sqrt{1-u^4}}\, du.$$

Outra alternativa seria utilizar a função inversa para resolver o problema:

$$\int_0^1 \sqrt{\operatorname{sen} x}\, dx = \int_{0,3}^1 \sqrt{\operatorname{sen} x}\, dx + 0,3\sqrt{\operatorname{sen} 0,3} - \int_0^{\sqrt{\operatorname{sen} 0,3}} \arcsin\, y^2\, dy$$

9.4 Intervalos de integração infinitos

Quando um ou os dois limites de integração de uma função são ∞, é necessário combinar o processo de integração numérica com uma manipulação algébrica adequada da função integranda, ou, alternativamente, determinar um valor que aproxime a região abaixo da curva da função a partir de um valor de x (ver [2]).

Considere a integral

$$\int_b^\infty \frac{1}{x + e^{-x} + x^2}\, dx \tag{9.32}$$

onde pode-se observar que $e^{-\infty} \approx 0$. Uma alternativa para se calcular (9.32) é notar que a curva da função

$$\frac{1}{x + x^2}$$

aproxima relativamente bem a função integranda em (9.32), como pode-se ver na figura 9.3. Nesse caso, pode-se escrever

$$\int_b^\infty \frac{1}{x + e^{-x} + x^2}\, dx < \int_b^\infty \frac{1}{x + x^2}\, dx = -\ln b + \ln(1+b). \tag{9.33}$$

Uma alternativa seria substituírmos e^{-x} em (9.32) por e^{-b}, já que esse valor poderia ser considerado não tão desprezível. Nesse caso, teríamos

$$\int_b^\infty \frac{1}{x + e^{-b} + x^2}\, dx = \left(csgn(\sqrt{4\, e^{-b} - 1})\pi - 2 \arctan(\frac{2\, b + 1}{\sqrt{4\, e^{-b} - 1}}) \right) \frac{1}{\sqrt{-\left(-4 + e^b\right) e^{-b}}} \tag{9.34}$$

Note que, nesse caso, calcular a antiderivada de $\frac{1}{x+e^{-b}+x^2}$ é bastante complicado, e, algumas vezes, a aproximação obtida com (9.33) é suficiente, como mostra o exemplo abaixo.

Exemplo 9.14 *Seja $b = 10$ em (9.32). Calculando a aproximação dessa integral através de (9.33), obtemos o valor $0,095310180$; utilizando (9.34), o valor obtido é .09531016670. Note que o erro relativo entre ambas aproximações é da ordem de 10^{-5}, o que pode não justificar o uso da segunda aproximação.*

9.5 Exercícios

Exercício 9.1 *Calcule a integral de $f(x) = \sqrt{6\, x + 5}$ no intervalo $[1, 9]$ com a fórmula dos trapézios considerando $h = 1$ e depois delimite o erro de truncamento para este caso.*

Exercício 9.2 *Determine h de tal forma que a regra dos trapézios forneça o valor de*

$$\int_0^1 e^{-x^2} dx$$

com um erro de truncamento menor do que 10^{-4}.

Exercício 9.3 *Calcule*

$$\int_6^{10} \log x \, dx$$

utilizando a fórmula de Simpson para 8 subintervalos e delimite o erro de truncamento.

Exercício 9.4 *Encontre n e h tal que o erro para a fórmula de Simpson seja menor do que 5×10^{-9} quando se quer aproximar*

$$\int_2^7 \frac{dx}{x}$$

Depois, faça o mesmo para a fórmula dos trapézios e compare os resultados.

Exercício 9.5 *Calcular uma aproximação de*

$$\int_0^1 \frac{1}{1+x^2} \, dx$$

pela regra de Simpson com exatidão crescente com no mínimo 5 DIGSE.

Exercício 9.6 *Usando quadratura de Gauss, calcule:*

$$\int_{-1}^1 x^2 \, dx \qquad \textit{com 4 pontos}$$

Exercício 9.7 *Usando quadratura de Gauss, calcule:*

$$\int_0^{10} e^{-x} \, dx \qquad \textit{com 2 pontos}$$

Depois, calcule o "erro exato" (diferença entre o valor da integral calculada com as regras do Cálculo e o valor obtido por quadratura) e use este valor para estimar o número mínimo de pontos necessários para calcular esta integral com a regra dos trapézios.

Exercício 9.8 *Sugira uma mudança de variável adequada para o cálculo da integral:*

$$\int_0^1 \frac{\operatorname{sen} x}{\sqrt{1-x^2}} \, dx.$$

Depois, encontre uma aproximação para o seu valor.

Exercício 9.9 *Utilize a regra de Simpson com exatidão crescente para calcular*

$$\int_0^1 \frac{\sqrt{x}}{4-x^2} \, dx$$

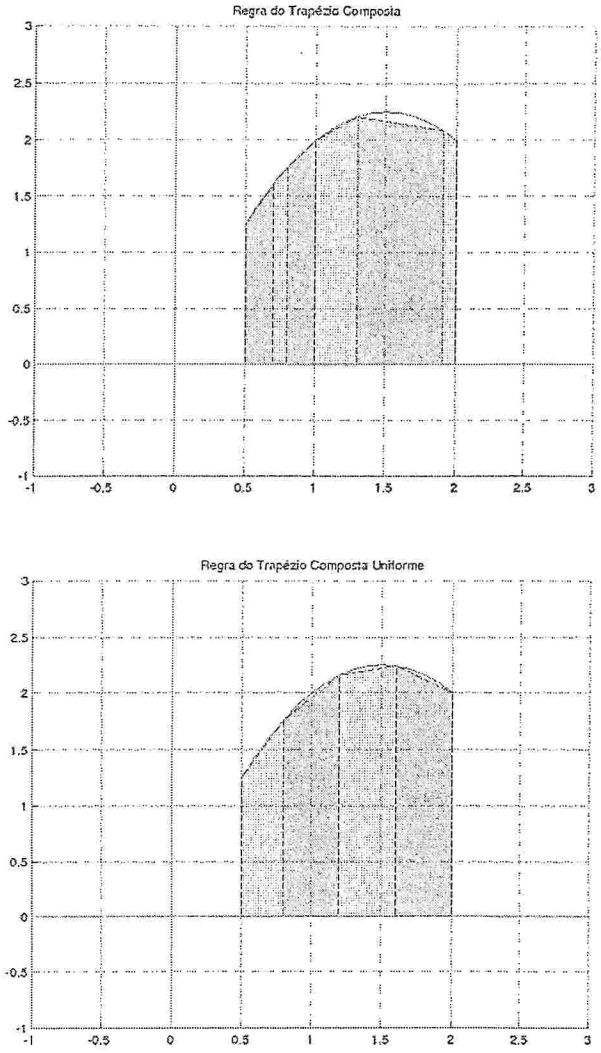

Figura 9.2: *A regra do trapézio composta: (a) subintervalos de qualquer tamanho, (b) subintervalos de tamanhos iguais.*

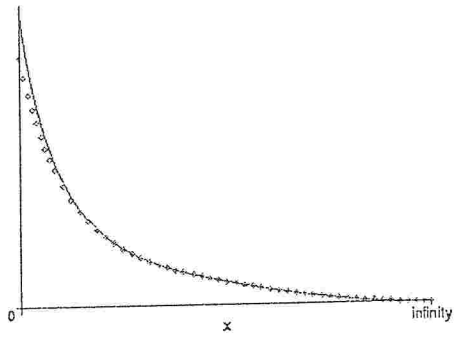

Figura 9.3: *Gráfico de* $\frac{1}{x+e^{-x}+x^2}$ (–) e $\int_b^\infty \frac{1}{x+x^2}$ ($\Diamond$).

Capítulo 10

Solução Numérica de Equações Diferenciais Ordinárias

10.1 Introdução

Equações diferenciais aparecem com grande freqüência em modelos que descrevem quantitativamente fenômenos em diversas áreas, como por exemplo mecânica dos fluidos, fluxo de calor, vibrações, reações químicas e nucleares, economia, biologia, etc. A motivação para a construção dos primeiros computadores foi ocasionada, em grande parte, pela necessidade de serem calculadas trajetórias balísticas de uma forma precisa e rápida. Hoje em dia, os computadores estão sendo muito empregados na solução de equações relacionadas com os foguetes balísticos, com a teoria de satélites artificiais, com o estudo de redes elétricas, curvaturas de vigas, estabilidade de aviões, teoria de vibrações e outras aplicações.

Exemplo 10.1 *Considere a equação*

$$\frac{dy}{dt} = 1 - e^{-t}. \tag{10.1}$$

Esta é uma equação diferencial porque envolve a derivada $\frac{dy}{dt}$ *de* $y = y(t)$. *Apenas a variável independente* t *aparece do lado direito da equação* (10.1). *Portanto, uma solução é a antiderivada de* $1 - e^{-t}$ *e as regras de integração podem ser empregadas para determinar* $y(t)$*:*

$$y(t) = t + e^{-t} + c, \tag{10.2}$$

onde c *é a constante de integração. Todas as funções em* (10.2) *são soluções de* (10.1) *porque satisfazem a condição* $y'(t) = 1 - e^{-t}$. *Na verdade, elas formam uma família de curvas.*[1]

Exemplo 10.2 *Considere a temperatura* $y(t)$ *de um objeto sob processo de resfriamento. A taxa de variação de temperatura do corpo está relacionada com a diferença de temperatura entre a sua temperatura e a do meio que o cerca. Este fenômeno pode ser expresso pela equação diferencial*

$$\frac{dy}{dt} = -k\,(y - A)$$

onde A *é a temperatura do meio,* y *é a temperatura do objeto no tempo* t *e* k *é uma constante positiva. O sinal negativo é necessário para garantir que* $\frac{dy}{dt}$ *será negativo quando a temperatura do corpo superar a temperatura do meio.*

[1] A variação do valor de c representa um movimento da curva solução para cima ou para baixo e é possível encontrar uma determinada curva que passe pelos pontos desejados.

Se a temperatura do objeto é conhecida no tempo $t = 0$, diz-se que esta é uma condição inicial e inclui-se esta informação na formulação do problema, que fica:

$$\frac{dy}{dt} = -k(y - A) \quad com \quad y(0) = y_0,$$

Pode-se usar a técnica de separação de variáveis para encontrar a solução

$$y = A + (y_0 - A)\, e^{-kt}.$$

Para cada escolha de y_0 a curva solução será diferente, como mostra a figura. Pode-se observar que, à medida que o tempo passa, a temperatura do objeto se aproxima da temperatura do meio. Se $y_0 < A$ então o objeto está sendo aquecido, e não resfriado.

Uma equação envolvendo uma relação entre uma função desconhecida e uma ou mais de suas derivadas é denominada *equação diferencial*. Assim, uma equação diferencial ordinária (que tem apenas uma variável independente) de ordem n tem a forma

$$y^{(n)} = f(x, y, y', y'' \ldots, y^{(n-1)}). \tag{10.3}$$

Sua solução é uma função $\phi(x)$ n vezes diferenciável em um intervalo determinado e que satisfaz (10.3), isto é,

$$\phi^{(n)} = f(x, \phi, \phi', \phi'' \ldots, \phi^{(n-1)}). \tag{10.4}$$

Exemplo 10.3

$$\frac{dy}{dx} = x + y \quad \textit{equação diferencial ordinária}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad \textit{equação diferencial parcial}$$

Diz-se que a ordem de uma equação diferencial é a ordem da mais alta derivada que aparece na equação. Uma equação diferencial é linear se a função e suas derivadas aparecem linearmente na equação.

Exemplo 10.4

$$x\,y' = x - y \qquad \textit{equação diferencial linear}$$

$$y'' + (1 - y^2)\, y' + y = 0 \qquad \textit{equação diferencial não linear}$$

Se, dada uma equação de ordem m, a função, assim como as suas derivadas até ordem $m - 1$ são especificadas em um mesmo ponto, então tem-se um problema de valor inicial – PVI. *Se, em problemas envolvendo equações diferenciais ordinárias de ordem $m \geq 2$, as m condições fornecidas para busca da solução única não são todas dadas em um mesmo ponto, então tem-se um* problema de valor de contorno – PVC.

Exemplo 10.5

$$\begin{cases} y'(x) &= y \\ y(0) &= 1 \end{cases} \qquad \textit{é um PVI}$$

$$\begin{cases} y^{(4)}(x) + k\,y(x) = q \\ y(0) = y'(0) = 0 \\ y(L) = y''(L) = 0 \end{cases} \qquad \textit{é um PVC}$$

Embora existam várias técnicas para solucionar, de forma aproximada, algumas classes selecionadas de equações diferenciais, a grande maioria das equações encontradas na prática não podem ser solucionadas analiticamente. Não existe, por exemplo, nenhuma "expressão fechada" para a solução de $y' = x^3 + y^2$ com $y(0) = 0$. Neste caso, os recursos disponíveis são os métodos numéricos, que aproximam a solução desejada.

Um procedimento numérico para calcular a solução de um dado PVI é um algoritmo para calcular os valores aproximados $y_0, y_1, y_2, \ldots, y_n, \ldots$ da solução $y = \phi(t)$ em um conjunto de pontos $t_0 < t_1 < t_2 < \ldots < t_n \ldots$, conforme a figura 10.1.

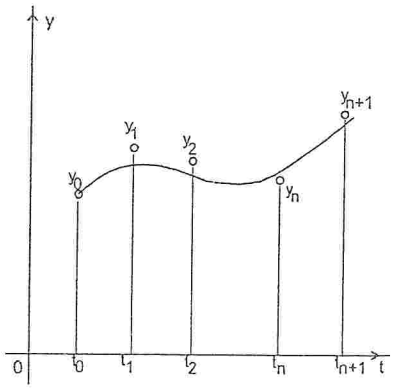

Figura 10.1: *Aproximação da solução em um PVI.*

10.2 Problema de Valor Inicial

Suponha o seguinte *problema de valor inicial* (PVI)

$$\begin{cases} x' &= f(t,x) \\ x(t_0) &= x_0 \end{cases} \tag{10.5}$$

onde x é uma função de t, com $x' = \frac{d}{dt}x(t)$. A função f dá a "inclinação" de x no ponto t. Por exemplo,

$$\begin{cases} x' &= \tan(t+3) \\ x(-1) &= 1 \end{cases} \tag{10.6}$$

A partir da equação (10.5), pretende-se determinar x em um intervalo contendo o ponto inicial t_0. Como a solução analítica do PVI (10.6) é $x(t) = \sec(t+3)$, podemos ver que $-\pi/2 < t+3 < \pi/2$, já que $\sec t$ é indefinida para $t = \pm\pi/2$.

Esse exemplo é *muito particular*, pois a sua solução analítica nos permite calcular valores para x. Tipicamente, no entanto, problemas da forma (10.5) *não tem solução analítica*, e métodos numéricos devem ser utilizados para obter-se uma solução *aproximada*, como nos exemplos a seguir:

1. A equação $y' = x^2 + y^2$ não tem solução elementar;
2. A equação $y'' \pm a(y')^2 + by = 0$, a qual descreve vibrações com amortecimento proporcional ao quadrado da velocidade, não pode ser solucionada de forma analítica;
3. Um problema de grande interesse histórico – a solução das equações diferenciais que governam o movimento de três corpos sujeitos às suas próprias atrações gravitacionais – também não tem solução analítica.

Em outras situações, é mais simples recorrer-se a uma solução numérica de um problema desse tipo, utilizando-a para obter valores de uma solução particular, como em

1. A equação $y'' = -xy$, sujeita a uma transformação de variáveis, pode ser resolvida em termos de funções de Bessel;
2. Uma tabela de integrais elípticas pode ser usada para resolver equações do tipo $\phi'' = -sen\phi$.

10.2.1 Existência da Solução

Cabe, agora, uma pergunta: será que todo e qualquer problema na forma (10.5) apresenta solução? A resposta é *não* e, mesmo assim, dependendo de certas considerações feitas a respeito de f, a solução, se existir, só será considerada *na vizinhança de* t_0. Por exemplo, considere

$$\begin{cases} x' &= 1+x^2 \\ x(0) &= 0 \end{cases} \tag{10.7}$$

A inclinação em $t = 0$ é 1 (i.e. $x'(0) = 1$). Como a inclinação é positiva, podemos dizer que $x(t)$ está *crescendo* perto de $t = 0$; logo, também $1 + x^2$ cresce. Ora, como x e x' crescem, para algum valor de t não haverá solução, qual seja, $x(t) = +\infty$; no entanto, o problema (10.7) apresenta como solução analítica $x(t) = \tan t$.

Vejamos então alguns teoremas que garantem a *existência* e a *unicidade* da solução (as provas dos mesmos podem ser verificadas em [10]).

Teorema 10.2.1 *Se f é contínua em um retângulo R centrado em (t_0, x_0), $R = \{(t,x) : |\, t-t_0\,| \leq \alpha, |\, x - x_0\,| \leq \beta\}$, então o problema (10.5) tem uma solução $x(t)$ para $|\, t - t_0\,| \leq \min(\alpha, \beta/M)$, onde M é o máximo de $|\, f(x,t)\,|$ no retângulo R.*

Teorema 10.2.2 *Se f e $\frac{\partial f}{\partial x}$ são contínuas no retângulo R, então o problema (10.5) tem uma solução única no intervalo $|\, t - t_0\,| < \min(\alpha, \beta/M)$.*

Teorema 10.2.3 *Se f é contínua na tira $a \leq t \leq b$, $-\infty < x < \infty$ e satisfaz a condição de Lipschitz em x,*

$$|\, f(t,x_1) - f(t,x_2\,| \leq L|\, x_1 - x_2\,|$$

então o problema (10.5) tem uma solução única em $[a,b]$.

10.2.2 Erros na solução numérica

Ao se aproximar numericamente a solução de uma equação diferencial – através de um processo de integração numérica – uma série de *erros* surgem, os quais podem ser classificados como:

Erro de truncamento local (ETL): é o erro existente em uma iteração da integração numérica ao substituirmos um processo *infinito* por um *finito*;

Erro de arredondamento local (EAL): é causado pela precisão finita do computador em uso;

Erro de truncamento global (ETG): é a acumulação dos ETL ao longo do processo de integração; porém, ele existiria mesmo que se utilizasse uma aritmética de precisão infinita, pois é inerente ao *método* e independente do computador utilizado;

Erro de arredondamento global (EAG): é a acumulação de todos os EAL;

Erro total (ETT): é a soma dos ETG e EAG.

A seguir, apresentaremos alguns dos diferentes métodos numéricos para se obter uma aproximação para a solução de (10.5).

10.2.3 Método da Série de Taylor

Usualmente, um método numérico para a solução de uma equação diferencial produz um conjunto de valores; em nosso caso, para o problema (10.5), teríamos os pares (t_0, x_0), (t_1, x_1), ..., (t_m, x_m). Isso nos mostra que a solução numérica é sempre *discreta*; obviamente, uma expressão analítica, contínua, pode ser obtida através da interpolação de uma "spline" ou outra função aos pontos calculados (ver Capítulo 7).

Mais uma vez, consideremos o problema (10.5); f é uma função de duas variáveis, e (t_0, x_0) um ponto *único* através do qual passa a curva solução. Essa *solução* é uma função $x(t)$ tal que

$$\frac{dx(t)}{dt} = f(t, x(t)) \forall t, \quad |t - t_0| \ll 1$$

com $x(t_0) = x_0$.

O *método da série de Taylor* consiste em obtermos uma expansão em x de $f(x,t)$, de acordo com a série de Taylor, até um determinado número de termos; uma vez obtida a expansão, integramo-la num intervalo $[t_0, t_1]$ e $[x_0, x_1]$. Considere, então, o problema

$$\begin{cases} x' &= \cos t - senx + t^2 \\ x(-1) &= 3 \end{cases} \tag{10.8}$$

Escrevendo a série de Taylor para x, temos

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \frac{h^4}{4!}x^{(4)}(t) + \ldots$$

de onde, para o problema (10.8), vem[2]

$$x'' = -sent - x' \cos x + 2t \tag{10.9}$$

$$x''' = -\cos t - x'' \cos x + (x')^2 senx + 2 \tag{10.10}$$

$$x^{(4)} = sent - x''' \cos x + 3x'x''senx + (x')^3 \cos x \tag{10.11}$$

Obviamente, cada termo de ordem superior a 4 será cada vez mais extenso; no entanto, pode-se observar que (10.11) é escrita em termos de (10.10) a qual, por sua vez, é escrita em termos de (10.9) a qual é escrita, também, em termos de x', dada no problema.

Utilizando apenas os termos acima, dizemos que o método de Taylor correspondente é de *quarta ordem* (de forma genérica, o método de Taylor de n-ésima ordem inclui todos os termos até $\frac{h^n}{n!}x^{(n)}(t)$). Os termos descartados (ordem superior a n) constituem o ETL. Na série de Taylor, o ETL é dado por

$$ETL_n = \frac{1}{(n+1)!}h^{n+1}x^{(n+1)}(t+\theta h), \quad 0 < \theta < 1 \tag{10.12}$$

O processo de integração consiste em se avaliar a série truncada de Taylor em diferentes pontos (t, x). Deve-se fixar o *intervalo de integração* em t, i.e. $t_0 \leq t \leq t_1$, bem como o *passo de integração*, h, tal que o número de iterações será $\frac{t_1 - t_0}{h}$. Daí, pode-se dizer que o ETG é, no mínimo, de ordem $O(h^n)$.

Pode-se afirmar que o método de Taylor é extremamente dependente do problema a ser resolvido, pois é necessário escrever explicitamente as derivadas (parciais) de $x(t)$. Dessa forma, o algoritmo mostrado a seguir é apenas um modelo para o método de Taylor de quarta ordem, o qual deve ser adaptado para cada problema específico.

[2]Note que ao diferenciar termos como $senx$ em relação a t, devemos ter $\frac{dsenx(t)}{dt}$, aplicando a regra da cadeia.

Algoritmo 10.2.1 *Método de Taylor de 4ª Ordem*

proc taylor_4(input: h, t_0, t_1, x_0; *output:* x)
  $n \leftarrow \frac{|t_1 - t_0|}{|h|}$
  $t \leftarrow t_0$
  $x \leftarrow x_0$
  for $k \leftarrow 0, 1, \ldots, n$ *do*
    % *Inclua as derivadas de* $f(x,t)$ *nas linhas abaixo*
    $x_1 \leftarrow \ldots$
    $x_2 \leftarrow \ldots$
    $x_3 \leftarrow \ldots$
    $x_4 \leftarrow \ldots$
    $x \leftarrow x + h\left(x_1 + \frac{h}{2}\left(x_2 + \frac{h}{3}\left(x_3 + \frac{h}{4}x_4\right)\right)\right)$
    $t \leftarrow t + h$
  endfor
endproc

Note que o algoritmo 10.2.1 avalia a expansão de Taylor utilizando um esquema de multiplicação aninhada, como no método de Horner (vide Seção 3.5). Além disso, o erro de truncamento *local* (i.e. a cada t) é da ordem de h^5. Se usarmos, por exemplo, um passo de integração $h = 10^{-2}$, esse erro será de 10^{-10} a cada iteração em t; é possível, para k muito grande, que esses erros, acumulados, contaminem o processo de integração numérica.

10.2.3.1 Vantagens e desvantagens

O método de Taylor exige a existência de derivadas parciais de f na região onde a curva solução passa no plano $t - x$. Veja que esta exigência não é necessária para a existência de solução. Além disso, cada derivada parcial deve ser individualmente codificada.

Como vantagens, o método é extremamente simples e, se for possível utilizar derivadas de maior ordem, a precisão do método é potencialmente alta.

10.2.4 Método de Euler

O método de Euler é uma simplificação do método de Taylor, e nada mais é do que um método de Taylor de 1ª ordem, i.e.

$$x_{t+h} = x(t) + hf(t,x) \tag{10.13}$$

O método de Euler, o qual pode ser expresso através do algoritmo 10.2.2, é bastante utilizado, por não exigir mais do que é expresso na definição de um problema do tipo (10.5), apesar de apresentar um ETL de ordem $O(h)$.

Cabe ressaltar que a primeira tentativa de resolução numérica de uma equação diferencial foi feita por *Euler*, por volta de 1768 D.C.

Algoritmo 10.2.2 *Método de Euler*

proc euler(input: h, t_0, t_1, x_0; *output:* x)
  $n \leftarrow \frac{|t_1 - t_0|}{|h|}$
  $t \leftarrow t_0$
  $x \leftarrow x_0$
  for $k \leftarrow 0, 1, \ldots, n$ *do*
    $x \leftarrow x + hf(t,x)$
    $t \leftarrow t + h$
  endfor
endproc

Exemplo 10.6 *Considere o problema*

$$\begin{aligned} y' &= 1 - t + 4y \qquad (10.14)\\ y(0) &= 1 \qquad (10.15)\end{aligned}$$

A equação (10.14) é do tipo diferencial linear de primeira ordem, sendo fácil verificar que a solução que cumpre a condição inicial (10.15) é

$$y = \phi(t) = \frac{1}{4}t - \frac{3}{16} + \frac{19}{16}e^{4t}.$$

Assim, com a fórmula de Euler e um incremento $h = 0,1$ é possível determinar um valor aproximado para a solução em $t = 0,2$ do PVI acima.

Neste caso, $f(t,y) = 1 - t + 4y$. Para usar a aproximação de Euler, calcula-se inicialmente $f_0 = f(0,1) = 5$. Então:

$$\begin{aligned} y_1 &= y_0 + h\,f(0,1)\\ &= 1 + (0,1)\,(5)\\ &= 1,5,\end{aligned}$$

Na etapa seguinte,

$$\begin{aligned} y_2 &= y_1 + h\,f(t_1,y_1)\\ &= 1,5 + (0,1)\,f(0,1\,,\,1,5)\\ &= 1,5 + (0,1)\,(6,9)\\ &= 2,19,\end{aligned}$$

Este resultado pode ser comparado com o valor exato de $\phi(0,2)$ que é $\phi(0,2) = 2,5053299$. Logo, o erro é aproximadamente $2,51 - 2,19 = 0,32$. Um erro desta grandeza (erro percentual de 12%) não é normalmente aceitável.

Sugestão: *Experimente refazer seus cálculos considerando espaçamentos progressivamente menores e observe o que acontece.*

10.2.5 Método de Heum

Este método introduz uma idéia nova para a construção de um algoritmo para solução do problema de valor inicial

$$\begin{aligned} y'(t) &= f(t,y(t))\\ y(t_0) &= y_0\end{aligned}$$

no intervalo $[a,b]$.

Para obter o ponto (t_1,y_1), usa-se o teorema fundamental do cálculo e integra-se $y'(t)$ sobre $[t_0,t_1]$:

$$\int_{t_0}^{t_1} y'(t)\,dt = y(t_1) - y(t_0), \qquad (10.16)$$

onde a antiderivada de $y'(t)$ é a função desejada, $y(t)$. Quando a equação (10.16) é resolvida para $y(t_1)$ o resultado é

$$y(t_1) = y(t_0) + \int_{t_0}^{t_1} f(t,y(t))\,dt. \qquad (10.17)$$

Agora, usando integração numérica, pode-se aproximar a integral definida em (10.17). A regra dos trapézios com passo $h = t_1 - t_0$ fornece

$$y(t_1) \approx y(t_0) + \frac{h}{2}\left[f(t_0,y(t_0)) + f(t_1,y(t_1))\right]. \qquad (10.18)$$

Note que esta fórmula ainda envolve o valor de $y(t_1)$, que não é conhecido. Por isto, usa-se a fórmula de Euler para estimar este valor. Substituindo o valor de $y(t_1)$ calculado pela fórmula de Euler na equação (10.18), a fórmula resultante é chamada de *método de Heum*:

$$y_1 = y_0 + \frac{h}{2}\left[f(t_0, y_0) + f(t_1, y_0 + h\, f(t_0, y_0))\right]. \tag{10.19}$$

Este processo é repetido, gerando uma seqüência de pontos que aproximam a solução $y = \phi(t)$. A cada passo, o método de Euler é usado como um preditor e a regra dos trapézios, como uma correção para que o valor final seja obtido.

A fórmula geral para o método de Heum é então dada por

$$p_{n+1} = y_n + h\, f(t_n, y_n) \tag{10.20}$$

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, p_{n+1})\right] \tag{10.21}$$

10.2.5.1 Erro de truncamento para o método de Heum

O erro de truncamento local para este método é da forma

$$|\, e_n \,| \leq \frac{h^3}{12} \max_{t \in I} \phi''(t) \tag{10.22}$$

onde $y = \phi(t)$ é a solução exata e o erro de truncamento global, ou seja, o erro acumulado depois de m passos, é da forma

$$|\, E_n \,| \leq C\, h^2$$

onde C é uma constante. Portanto, quando o passo h é reduzido por um fator de $\frac{1}{2}$, pode-se esperar que o erro de truncamento global seja reduzido por um fator de $\frac{1}{4}$.

Observação: A fórmula de Heum é um exemplo de um método em dois estágios: calcula-se primeiro $y_n + h\, f_n$ pela fórmula de Euler e depois utiliza-se este resultado para calcular y_{n+1} com a equação (10.21). O aprimoramento da equação (10.21) em relação à fórmula de Euler está no fato de que o erro de truncamento local da equação (10.21) é $O(h^3)$, ao passo que, para o método de Euler, este é $O(h^2)$. Note que esta melhoria de precisão é conseguida com maior esforço computacional, pois é preciso estimar $f(t, y)$ duas vezes a fim de passar de t_n para t_{n+1}.

Se $f(t, y)$ depender exclusivamente de t e não de y, a resolução da equação diferencial $y' = f(t, y)$ se reduz à integração de $f(t)$. Neste caso, o método de Heum reduz-se a

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n) + f(t_{n+1})\right],$$

que é a regra dos trapézios para integração numérica.

Exemplo 10.7 *Use o método de Heum com $h = 1$ para aproximar $y(2)$ para o problema*

$$\begin{aligned} y' &= \frac{t - y}{2} \\ y(0) &= 1 \end{aligned}$$

O primeiro passo é calcular p_1 com o método de Euler:

$$\begin{aligned} p_1 &= y_0 + h\, f(t_0, y_0) \\ &= 1 + 1\,\frac{0 - 1}{2} = 0,5 \end{aligned}$$

Agora, utiliza-se efetivamente a fórmula de Heum para calcular y_1, que é uma aproximação para a solução em $t = 1$:

$$\begin{aligned} y_1 &= y_0 + \frac{h}{2}\left[f(t_0, y_0) + f(t_1, p_1)\right] \\ &= 1 + \frac{1}{2}\,(-0,5 + 0,25) = 0,875 \end{aligned}$$

Para aproximar $y(2)$, calcula-se p_2 e y_2:

$$\begin{aligned} p_2 &= y_1 + h f(t_1, y_1) \\ &= 0,875 + 1\,\frac{1 - 0,875}{2} = 0,9375 \end{aligned}$$

e

$$\begin{aligned} y_2 &= y_1 + \frac{h}{2}\,[f(t_1, y_1) + f(t_2, p_2)] \\ &= 0,875 + \frac{1}{2}\,(0,0625 + 0,53125) = 1,171875 \end{aligned}$$

10.2.6 Métodos de Runge-Kutta

Os *métodos de Runge-Kutta* são similares ao método da série de Taylor, com a vantagem de não necessitarem das derivadas de ordem superior a 1, Vejamos isso através do *método de Runge-Kutta de segunda ordem.*

Escrevendo a série de Taylor para $x(t+h)$, vem

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \ldots \tag{10.23}$$

e, valendo-nos do PVI (10.5), temos, usando a regra de cadeia,

$$\begin{aligned} x'(t) &= f \\ x''(t) &= f_t + f_x x' = f_t + f_x f \\ x'''(t) &= f_{tt} + f_{tx} f + (f_t + f_x f) f_x + f(f_{xt} + f_{xx} f) \end{aligned}$$

onde o subscrito indica derivação parcial em relação àquela variável. Daí, os primeiros três termos de (10.23) podem ser reescritos como

$$\begin{aligned} x(t+h) &= x + hf + \frac{1}{2}h^2(f_t + f f_x) + O(h^3) \\ &= x + \frac{1}{2}hf + \frac{1}{2}h(f + hf_t + hf f_x) + O(h^3) \end{aligned} \tag{10.24}$$

onde $x \equiv x(t)$ e $f \equiv f(t,x)$. Podemos eliminar de (10.24) os termos envolvendo as derivadas parciais f_t, f_x, usando os primeiros termos da série de Taylor em duas variáveis,

$$f(t+h, x+hf) = f + hf_t + hf f_x + O(h^2)$$

de onde (10.24) pode ser reescrita como

$$x(t+h) = x + \frac{1}{2}hf + \frac{1}{2}hf(t+h, x+hf) + O(h^3)$$

e, descartando $O(h^3)$, escrevemos

$$x(t+h) = x(t) + \frac{1}{2}(F_1 + F_2), \quad F_1 = hf(t,x), \quad F_2 = hf(t+h, x+F_1) \tag{10.25}$$

a qual é a fórmula para o *método de Runge-Kutta de segunda ordem*, também chamado de *método de Heum.*

De forma geral, temos

$$x(t+h) = x + w_1 hf + w_2 hf(t+\alpha h, x+\beta hf) + O(h^3) \tag{10.26}$$

onde w_1, w_2, α e β são parâmetros a escolher.

Reescrevendo (10.26) usando a série de Taylor em duas variáveis, temos

$$x(t+h) = x + w_1 hf + w_2 h(f + \alpha h f_t + \beta h f f_x) + O(h^3) \tag{10.27}$$

e, comparando (10.24) com (10.27), vemos que as seguintes condições devem ser impostas:

$$\begin{cases} w_1 + w_2 &= 1 \\ w_2 \alpha &= \frac{1}{2} \\ w_2 \beta &= \frac{1}{2} \end{cases} \tag{10.28}$$

e, em (10.25), temos $w_1 = w_2 = 1/2$, $\alpha = \beta = 1$.

10.2.6.1 Método modificado de Euler

O *método modificado de Euler* é obtido com $w_1 = 0$, $w_2 = 1$ e $\alpha = \beta = 1/2$,

$$x(t+h) = x(t) + F_2, \quad F_1 = hf(t,x), \quad F_2 = hf\left(t + \frac{1}{2}h, x + \frac{1}{2}F_1\right) \tag{10.29}$$

10.2.6.2 Método de Runge-Kutta de 4^a Ordem

O *método de Runge-Kutta de 4^a ordem* pode ser escrito como

$$\begin{aligned} x(t+h) &= x(t) + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4) \\ F_1 &= hf(t,x) \\ F_2 &= hf\left(t + \frac{1}{2}h, x + \frac{1}{2}F_1\right) \\ F_3 &= hf\left(t + \frac{1}{2}h, x + \frac{1}{2}F_2\right) \\ F_4 &= hf(t+h, x+F_3) \end{aligned} \tag{10.30}$$

10.2.6.3 Erros do método de Runge-Kutta

Mais uma vez, nos deparamos com um ETL, o qual é da ordem de $O(h^5)$ para o método de Runge-Kutta de 4^a ordem. Vejam que no primeiro passo, um valor $\tilde{x}(t_0 + h)$ é calculado; existe também um valor $x(t_0 + h)$, o qual é o valor *exato* (e desconhecido), tal que o erro $e = x - \tilde{x}$ é Ch^5, para $h \ll 1$. O valor de C independe de h mas depende de t_0 e de x.

Para estimar Ch^5, assumamos que C *não* muda quando h é somado a t_0; chamemos de u o valor da solução em $t_0 + h$ e de v o valor da solução em $t_0 + \frac{h}{2} + \frac{h}{2}$. Temos, então,

$$\begin{aligned} x(t_0 + h) &= v + Ch^5 \\ x(t_0 + h) &= u + 2C\left(\frac{h}{2}\right)^5 \end{aligned}$$

e, subtraindo v de u, obtemos

$$Ch^5 = \frac{u - v}{1 - 2^{-4}} \tag{10.31}$$

Com isso, é possível estimar o ETL, calculando-se $|u - v|$, e verificando se ele encontra-se abaixo de uma tolerância pré-especificada (por exemplo, 10^{-5}). Se não estiver, então o passo h pode ser reduzido (normalmente pela metade); caso contrário, se o ETL é *muito* menor do que aquela tolerância, h pode ser aumentado (multiplicando-o por dois, normalmente).

10.2.6.4 Avaliação da Função versus Ordem do Método Runge-Kutta

A tabela 10.1 mostra que, ao se aumentar a ordem do método, o número de vezes que a função deve ser avaliada cresce rapidamente. Essa é a principal razão pela qual não se utilizam métodos de Runge-Kutta de ordem muito grande.

Ordem	1	2	3	4	4	5	6	6
Avaliação	1	2	3	4	5	6	7	8

Tabela 10.1: *Número de vezes que $f(t,x)$ deve ser avaliada nos métodos de Runge-Kutta.*

i	a_i	$a_i - b_i$	c_i	d_{i1}	d_{i2}	d_{i3}	d_{i4}	d_{i5}
1	$\frac{16}{135}$	$\frac{1}{360}$	0	0				
2	0	0	$\frac{1}{4}$	$\frac{1}{4}$				
3	$\frac{6656}{12825}$	$-\frac{128}{4275}$	$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$			
4	$\frac{28561}{56430}$	$-\frac{2197}{75240}$	$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$		
5	$-\frac{9}{50}$	$\frac{1}{50}$	1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	
6	$\frac{2}{55}$	$\frac{2}{55}$	$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$

Tabela 10.2: *Coeficientes do método Runge-Kutta-Fehlberg.*

10.2.6.5 Método Adaptativo de Runge-Kutta-Fehlberg

Em 1969, Fehlberg propôs um método que permite ajustar o passo de integração num método de Runge-Kutta, de forma adaptativa. Esse método baseia-se na combinação do método de Runge-Kutta de quarta ordem com cinco avaliações, com o método de Runge-Kutta de quinta ordem com seis avaliações. À primeira vista, tal método – abreviado por RKF – é desvantajoso com relação ao método clássico; porém, ele combina as constantes envolvidas nos dois métodos de Runge-Kutta utilizados, de forma a obter duas fórmulas, de diferentes ordens, as quais envolvem valores de $f(t,x)$ avaliadas nos *mesmos* pontos.

O método RKF é de quinta ordem e obtém duas aproximações diferentes para a solução, $x(t+h)$ e $\bar{x}(t+h)$, dadas por

$$x(t+h) = x(t) + \sum_{i=1}^{6} a_i F_i \tag{10.32}$$

$$\bar{x}(t+h) = x(t) + \sum_{i=1}^{6} b_i F_i \tag{10.33}$$

$$F_i = hf\left(t + c_i h, x + \sum_{j=1}^{i-1} d_{ij} F_j\right), \quad i = 1, 2, \ldots, 6 \tag{10.34}$$

onde os valores dos coeficientes presentes nas equações (10.32)-(10.33) são dados na tabela 10.2.

A equação (10.32) é de quinta ordem e a equação (10.33) é de quarta ordem. O ETL do método RKF é dado pela diferença entre ambas,

$$e = x(t+h) - \bar{x}(t+h) = \sum_{i=1}^{6} (a_i - b_i) F_i \tag{10.35}$$

e, portanto, e pode ser usado para se monitorar o comportamento do algoritmo.

O algoritmo adaptativo procura ajustar o valor do passo, h, sempre que o erro, e, tornar-se maior do que uma tolerância δ, pré-especificada. Procura-se, então, uma solução $\hat{x}$ no intervalo $t_0 \leq t \leq t_1$, com $\hat{x}(t_0) = x_0$. O passo do método pode ser alterado sempre que o erro exceder ao ETL (Ch^5); nesse caso, devemos reduzir o passo (pela metade, por exemplo). Note que, sempre que o passo for reduzido, deve-se descartar os últimos valores de t e x.

Por outro lado, a escolha de h pode ter sido muito conservadora, i.e. o valor de h é pequeno demais para aquele problema; nesse caso, podemos aumentar h, dobrando-o, sempre que $C(2h)^5 < \frac{\delta}{4}$, ou seja, $Ch^5 < \frac{\delta}{128}$.

O algoritmo 10.2.3 ilustra o método de Runge-Kutta-Fehlberg, incorporando o controle do passo apresentado.

Algoritmo 10.2.3 *Método de Runge-Kutta-Fehlberg*

proc runge_kutta_fehlberg(input: h, t_0, t_1, x_0, δ; *output:* x)
  % *Inicializa* a_i, $a_i - b_i$, c_i, d_i
  % *de acordo com a tabela 10.2*
  $n \leftarrow \frac{|t_1 - t_0|}{|h|}$
  $t \leftarrow t_0$
  $x \leftarrow x_0$
  $k \leftarrow 0$
  while $k \leq n$ *do*
    $F_i \leftarrow hf\left(t + c_i h, x + \sum_{j=1}^{i-1} d_{ij} F_j\right), \quad i = 1, 2, \ldots, 6$
    $\bar{x} \leftarrow x$
    $x \leftarrow x + \sum_{i=1}^{6} a_i F_i$
    $e \leftarrow \sum_{i=1}^{6} (a_i - b_i) F_i$
    $\Delta \leftarrow t_1 - t$
    if $|\Delta| \leq |h|$ *then*
      $h \leftarrow \Delta$
    endif
    $\bar{t} \leftarrow t$
    if $|e| \geq \delta$ *then*
      $h \leftarrow \frac{h}{2}$
      $x \leftarrow \bar{x}$
      $t \leftarrow \bar{t}$
      $n \leftarrow n + 1$
    endif
    if $|e| < \frac{\delta}{128}$ *then*
      $h \leftarrow 2h$
    endif
  endwhile
endproc

10.2.7 Métodos de passo múltiplo

Os métodos da série de Taylor e de Runge-Kutta são chamados de métodos de *passo simples*, pois apenas $x(t)$ é utilizado para se obter $x(t+h)$.

Suponhamos agora que, para um conjunto de nós $t_0, t_1, \ldots, t_n$, tenhamos calculado os valores $f(t_0, x)$, $f(t_1, x)$, $\ldots$, $f(t_n, x)$. Note que o espaçamento entre os nós não é necessariamente o mesmo. Chamando de $x(t)$ a solução ao PVI (10.5), então

$$\int_{t_n}^{t_{n+1}} x'(t)dt = x(t_{n+1}) - x(t_n)$$

de onde podemos escrever

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t))dt \tag{10.36}$$

Para aproximar a integral em (10.36), usaremos

$$\int_{t_n}^{t_{n+1}} f(t, x(t))dt \approx h(Af_n + Bf_{n-1} + Cf_{n-2} + Df_{n-3} \tag{10.37}$$

onde $f_i \equiv f(t_i, x(t_i))$, e A, B, C e D são coeficientes obtidos exigindo-se que (10.37) seja exata sempre que o integrando seja um polinômio de grau menor ou igual a 3. Usando a base de Newton

para representar o espaço polinomial Π^3,

$$\begin{cases} p_0(t) &= 1 \\ p_1(t) &= t \\ p_2(t) &= t(t+1) \\ p_3(t) &= t(t+1)(t+2) \end{cases}$$

e tomando $t_0 = 0$, $h = 1$, podemos aproximar o membro esquerdo de (10.37) por

$$\int_0^1 p_i(t)dt = Ap_i(0) + Bp_i(-1) + Cp_i(-2) + Dp_i(-3), \quad i = 0, 1, \ldots n$$

o que nos leva a um sistema de cinco equações a cinco variáveis:

$$\begin{cases} Ap_0(0) + Bp_0(-1) + Cp_0(-2) + Dp_0(-3) &= \int_0^1 p_0(t)dt \\ Ap_1(0) + Bp_1(-1) + Cp_1(-2) + Dp_1(-3) &= \int_0^1 p_1(t)dt \\ \qquad\qquad\qquad\qquad\qquad\qquad \ddots \quad \vdots \quad \ddots & \\ Ap_3(0) + Bp_3(-1) + Cp_3(-2) + Dp_3(-3) &= \int_0^1 p_3(t)dt \end{cases} \tag{10.38}$$

Como os p_i são conhecidos, o termo independente do sistema tem seus elementos perfeitamente determinados:

$$\begin{aligned} \int_0^1 1\,dt &= 1 \\ \int_0^1 t\,dt &= \frac{1}{2} \\ \int_0^1 t(t+1)\,dt &= \frac{5}{6} \\ \int_0^1 t(t+1)(t+2)\,dt &= \frac{9}{4} \end{aligned}$$

Podemos, portanto, escrever o sistema (10.38) na forma matricial

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ & -1 & -2 & -3 \\ & & 2 & 6 \\ & & & -6 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{5}{6} \\ \frac{9}{4} \end{bmatrix}$$

cuja solução é

$$A = \frac{55}{24},\ B = -\frac{59}{24},\ C = \frac{37}{24},\ D = -\frac{9}{24}.$$

Assim, podemos escrever (10.37) como

$$\int_{t_n}^{t_{n+1}} f(t, x(t))dt \approx \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

e, substituindo em (10.36), temos

$$x_{n+1} = x_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \tag{10.39}$$

com $t_i = t_0 + ih, 0 \leq i \leq n$, a qual é conhecida como *fórmula de Adams-Bashforth de* 4^a *ordem.*

De forma a aumentar a exatidão da solução, a fórmula de Adams-Bashforth é usada em conjunto com outras fórmulas, de maior ordem. Por exemplo, se incluirmos o termo f_{n+1} na aproximação

para a integral em (10.36), e descartarmos o termo f_{n-4} , podemos obter, de forma semelhante, a *fórmula de Adams-Moulton de* 4^a *ordem*,

$$x_{n+1} = x_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \tag{10.40}$$

Veja que a equação (10.40) não pode ser usada para se obter x_{n+1} a partir de x_n, pois f_{n+1} é calculada em x_{n+1}. Mas, se usarmos a fórmula de Adams-Bashforth (10.39) para *predizer* um valor $\tilde{x}_{n+1}$ para x_{n+1}, e usarmos (10.40) para *corrigir* $\tilde{x}_{n+1}$ (com $f_{n+1} \equiv f(t_{n+1}, \tilde{x}_{n+1})$, obtemos um algoritmo altamente eficaz, chamado de *"previsor-corretor"*. Inicialmente, é necessário obter valores para x_1, x_2, x_3 e x_4 – usualmente através do método de Runge-Kutta – e, então, estimar x_{n+1} através de (10.39), corrigindo essa estimativa com (10.40).

Exemplo 10.8 *Considere o problema de valor inicial*

$$\begin{aligned} y' &= 1 - t + 4\,y \\ y(0) &= 1 \end{aligned}$$

Determine um valor aproximado da solução $y(t)$ *com um incremento* $h = 0,1$ *em* $t = 0,4$. *Use as fórmulas de quarta ordem de Adams-Bashforth, de Adams-Moulton e de predição-correção.*
Solução: *Como dados iniciais, usa-se os valores iniciais* y_1, y_2 *e* y_3 *determinados com o auxílio do método de Runge-Kutta de quarta-ordem. Em seguida, calculando os valores correspondentes de* $f(t, y)$, *obtém-se*

$$\begin{aligned} y_0 &= 1 & f_0 &= 5 \\ y_1 &= 1,6089333 & f_1 &= 7,3357332 \\ y_2 &= 2,5050062 & f_2 &= 10,820025 \\ y_3 &= 3,8294145 & f_3 &= 16,017658 \end{aligned}$$

1. *Usando a fórmula de Adams-Bashforth, determina-se que* $y_4 = 5,7836305$.

2. *A fórmula de Adams-Moulton leva à equação*

$$y_4 = 4,9251275 + 0,15\,y_4,$$

   de onde $y_4 = 5,7942676$.

3. *Finalmente, usando o resultado da fórmula de Adams-Bashforth como preditor, pode-se usar a fórmula (10.40) como corretor. O valor do preditor,* $y_4 = 5,7836305$, *leva a* $f_4 = 23,734522$ *e, de acordo com a equação (10.39), o valor corrigido de* y_4 *é* $5,7926721$.

Considerando que o valor da solução exata no ponto $t = 0,4$ *seja* $\phi(0,4) = 5,7942260$, *o método de Adams-Bashforth, embora seja o mais simples e rápido (já que envolve o uso de uma única fórmula explícita), é o menos preciso. O uso da fórmula de Adams-Moulton como correção aumenta o número de cálculos necessários, mas o método continua a ser explícito. Neste problema, o erro no valor corrigido de* y_4 *(*$y_4 = 5,7926721$*) é reduzido aproximadamente sete vezes em relação ao erro no valor do preditor (*$y_4 = 5,7836305$*). O método de Adams-Moulton sozinho (*$y_4 = 5,7942676$*) é o que fornece o resultado mais preciso, com erro 40 vezes menor que o erro associado ao método preditor-corretor. É necessário não esquecer, porém, que o método de Adams-Moulton é implícito, o que significa que é preciso resolver uma equação em cada passo. Neste problema em questão, a equação é linear, de modo que a solução não é difícil de encontrar. Entretanto, em outros problemas esta parte do processo pode ser muito mais demorada.*

10.2.7.1 Convergência, Estabilidade e Consistência

Todo método de passo múltiplo pode ser descrito por uma equação do tipo

$$a_k x_n + a_{k-1}x_{n-1} + \ldots + a_0 x_{n-k} = h(b_k f_n + b_{k-1}f_{n-1} + \ldots + b_0 f_{n-k}) \tag{10.41}$$

onde x_0, x_1, ..., x_{k-1} são obtidos por algum outro método, e (10.41) é usada com $n = k$, $k+1$, ... A equação (10.41) é dita ser *implícita* se $b_k \neq 0$, pois x_n aparecerá em ambos os lados da igualdade; caso contrário, o método é dito *explícito*.

Um método de passo múltiplo definido por (10.41) é dito ser *convergente* se

$$\lim_{h\to 0} x(h,t) = x(t) \tag{10.42}$$

com t fixo, e h livre, para todo t num intervalo $t_0 \leq t \leq t_m$, desde que os valores iniciais satisfaçam a mesma equação e f satisfaça o teorema básico de existência de solução (10.2.1), i.e.

$$\lim_{h\to 0} x(h, t_0 + nh) = x_0, \quad 0 \leq n < k \tag{10.43}$$

Para se analisar a *estabilidade* e a *consistência* de um método de passo múltiplo, utilizamos dois polinômios, associados à equação (10.41):

$$\begin{cases} p(z) = a_k z^k + a_{k-1}z^{k-1} + \ldots + a_0 \\ q(z) = b_k z^k + b_{k-1}z^{k-1} + \ldots + b_0 \end{cases} \tag{10.44}$$

Então, as condições necessárias para a estabilidade e consistência podem ser escritas como:

Estabilidade: O método é dito *estável* se todas as raízes de p estão contidas em um disco de raio $|z| \leq 1$ e se cada raiz de módulo 1 é simples.

Consistência: O método é dito *consistente* se $p(1) = 0$ e $p'(1) = q'(1)$.

O teorema a seguir estabelece que a convergência de um método de passo múltiplo depende da estabilidade e da consistência.

Teorema 10.2.4 *Um método de passo múltiplo conforme a equação (10.41) é convergente se e somente se ele é* estável e consistente.

A prova pode ser consultada em [10].

De posse desses resultados, podemos verificar se um método é convergente, como mostra o exemplo a seguir.

Exemplo 10.9 *O* método de Milne,

$$x_n - x_{n-2} = h\left(\frac{1}{3}f_n + \frac{4}{3}f_{n-1} + \frac{1}{3}f_{n-2}\right) \tag{10.45}$$

é um método implícito, *caracterizado por*

$$\begin{cases} p(z) = z^2 - 1 \\ q(z) = \frac{1}{3}z^2 + \frac{4}{3}z + \frac{1}{3} \end{cases}$$

cujas raízes de $p(z)$ são $+1$ e -1 (ambas simples). Como $p'(z) = 2z$, $p'(1) = 2$ e $q(1) = 2$, o método é estável e consistente, *logo é* convergente.

10.2.7.2 Erros de truncamento

Suponha que a equação (10.41) foi utilizada para se calcular x_n, e que $x_{n-1}, x_{n-2}, \ldots$ são *exatos*, i.e. $x_i = x(t_i)$ para $i < n$, onde $x(t)$ é a solução da equação diferencial. Então, o ETL é definido como $e = x(t_n) - x_n$. Este erro *não* é devido a erros de arredondamento mas sim devido à formulação (10.41).

Podemos definir um operador funcional linear L, correspondente a (10.41), dado por

$$Lx = \sum_{i=0}^{k} (a_i x(ih) - hb_i x'(ih)) \tag{10.46}$$

assumindo, por simplicidade, $k = n$ e $t = 0$. A operação Lx pode ser aplicada a qualquer função x diferenciável. Representando x por uma expansão de Taylor em $t = 0$, L pode ser expresso por

$$Lx = d_0 x_0 + d_1 h x'(0) + d_2 h x''(0) + \ldots \tag{10.47}$$

onde os coeficientes d_i são obtidos rearranjando os termos h, ao substituirmos na equação (10.46) as expressões para x e x', na forma de Taylor:

$$\begin{aligned} x(ih) &= \sum_{j=0}^{\infty} \frac{(ih)^j}{j!} x^{(j)}(0) \\ x'(ih) &= \sum_{j=0}^{\infty} \frac{(ih)^j}{j!} x^{(j+1)}(0) \end{aligned}$$

resultando, então, em

$$\begin{cases} d_0 = \sum_{i=0}^{k} a_i \\ d_1 = \sum_{i=0}^{k} (ia_i - b_i) \\ d_2 = \sum_{i=0}^{k} \left(\frac{i^2}{2} a_i - ib_i \right) \\ \ddots \\ d_j = \sum_{i=0}^{k} \left(\frac{i^j}{j!} a_i - \frac{i^{j-1}}{(j-1)!} b_i \right), \quad j = 1, 2, \ldots \end{cases} \tag{10.48}$$

De posse desses coeficientes, podemos estabelecer a ordem do erro de truncamento local.

Teorema 10.2.5 *Se (10.41) é de ordem m, $x \in \mathbb{C}^{m+2}$, $\frac{\partial f}{\partial x}$ é contínua, e $x_{n-1}, x_{n-2}, \ldots$ são exatos, então*

$$x(t_n) - x_n = \frac{d_{m+1}}{a_k} h^{m+1} x^{(m+1)} t_{n-k} + O\left(h^{m+2}\right)$$

e o ETL é, portanto, de ordem $O\left(h^{m+1}\right)$.

10.2.7.3 Erros de truncamento globais

Suponha que todos os cálculos foram efetuados com precisão infinita (sem erros de arredondamento), e que em t_n temos calculado o valor de x_n, o qual difere da solução exata $x(t_n)$. Note que x_n é diferente de $x(t_n)$ pois ele é obtido por uma aproximação a uma série de Taylor.

O ETG é definido como $x(t_n) - x_n$, e ele não é simplesmente a *soma* de todos os erros locais. Como na iteração usamos o valor x_{n-1} para aproximar x_n, e x_{n-1} tem um erro, então o processor numérico está, na verdade, seguindo uma curva solução "errada". O que acontece, então, quando duas diferentes condições iniciais são utilizadas?

Consideremos o PVI

$$\begin{cases} x' = f(t, x) \\ x(0) = s \end{cases} \tag{10.49}$$

com $f_x = \frac{\partial f}{\partial x}$ contínua e $f_x(t, x) \leq \lambda$ em $0 \leq t \leq T$, $\infty < x < \infty$.

A solução de (10.49) é uma função em t, dependente do valor inicial s, e a denotamos então como $x(t,s)$; definimos, ainda, $u(t) = \frac{\partial}{\partial s}x(t,s)$. Podemos obter uma equação diferencial para u diferenciando (10.49) em relação a s,

$$\begin{cases} u' = f_x(t,x)u \\ u(0) = 1 \end{cases} \tag{10.50}$$

a qual é chamada de *equação variacional.* O exemplo a seguir ilustra como obter u.

Exemplo 10.10 *Determine u explicitamente no PVI*

$$\begin{cases} x' = x^2 \\ x(0) = s \end{cases}$$

Solução: *A derivada de f em relação a x é $f_x = 2x$, logo a equação variacional é*

$$\begin{cases} u' = 2xu \\ u(0) = 1 \end{cases}$$

A solução do PVI é $x(t) = s(1-st)^{-1}$, logo

$$\begin{cases} u(t)' = 2s(1-st)^{-1}u(t) \\ u(0) = 1 \end{cases}$$

de onde $u(t) = 1(1-st)^{-2}$.

Os teoremas que seguem permitem estabelecer a ordem do ETG.

Teorema 10.2.6 *Se $f_x \leq \lambda$, então a solução da equação variacional satisfaz $|\,u(t)\,| \leq e^{\lambda t}$, para $t \geq 0$.*

Prova: *Por (10.50), vem*

$$\frac{u'}{u} = f_x = \lambda - \alpha(t)$$

onde $\alpha(t) \geq 0$. Integrando, vem

$$\log|\,u\,| = \lambda t - \int_0^t \alpha(\tau)\,d\tau$$

e, como $t \geq 0$, a integral na equação acima é maior ou igual a zero; conseqüentemente, $\log|\,u\,| \leq \lambda t$. Como a função exponencial é crescente para $t \geq 0$, $|\,u\,| \leq e^{\lambda t}$. $\diamond$

Teorema 10.2.7 *Se a equação (10.49) é resolvida com valores iniciais s e $s+\delta$, as curvas solução em t diferem de, no máximo, $|\,\delta\,|e^{\lambda t}$.*

Prova: *Usando o teorema do valor médio e o teorema 10.2.6,*

$$|\,x(t,s) - x(t,s+\delta)\,| = \left|\frac{\delta}{\delta s}x(t,s+\theta\delta)\right| |\,\delta\,| = |\,u(t)\,||\,\delta\,| \leq |\,\delta\,|e^{\lambda t} \diamond$$

Teorema 10.2.8 *Se os erros de truncamento locais em t_1, t_2, ..., t_n não excedem δ em magnitude, então o ETG em t_n não excederá $\delta(e^{n\lambda h}-1)(e^{\lambda h}-1)^{-1}$.*

Prova: *Sejam δ_1, δ_2, ... os ETLs associados aos pontos t_1, t_2, ... Ao calcular x_2, havia um erro δ_1 na condição inicial e, pelo teorema 10.2.7, o efeito deste erro em t_2 é de, no máximo, $|\,\delta_1\,|e^{\lambda h}$, ao qual é adicionado o ETL em t_2. Logo, o ETG nesse nó é de, no máximo, $|\,\delta_1\,|e^{\lambda h} + \delta_2$; por analogia, em t_3 teremos $(|\,\delta_1\,|e^{\lambda h} + \delta_2)e^{\lambda h} + |\,\delta_3\,|$ e, então*

$$\sum_{k=1}^{n} |,\delta_k\,|e^{(n-k)\lambda h} \leq \delta \sum_{k=0}^{n-1} e^{k\lambda h} = \delta\frac{e^{n\lambda h}-1}{e^{\lambda h}-1} \quad \diamond$$

Teorema 10.2.9 *Se o ETL é de ordem $O\left(h^{m+1}\right)$, então o ETG é de ordem $O\left(h^m\right)$.*

Prova: *No teorema 10.2.8, seja δ de ordem $O\left(h^{m+1}\right)$. Como $e^z - 1$ é de ordem $O(z)$ e $nh = t$, temos uma redução de uma unidade na ordem, usando a fórmula no teorema 10.2.8.* $\diamond$

10.2.8 Sistemas de Equações Diferenciais Ordinárias

Um sistema de equações diferenciais ordinárias é expresso como

$$\begin{cases} x_1' = f_1(t, x_1, x_2, \ldots, x_n) \\ x_2' = f_2(t, x_1, x_2, \ldots, x_n) \\ \ddots \\ x_n' = f_n(t, x_1, x_2, \ldots, x_n) \end{cases} \tag{10.51}$$

onde n funções x_1, x_2, ..., x_n devem ser determinadas. Elas são funções da variável independente t e $x_i' = \frac{d}{dt}x_i$. Como exemplo, considere

$$\begin{cases} x' = x + 4y - e^t \\ x' = x + y + 2e^t \end{cases} \tag{10.52}$$

cuja solução geral é

$$\begin{cases} x = 2ae^{3t} - 2be^{-t} - 2e^t \\ y = ae^{3t} + be^{-t} + \frac{1}{4}e^t \end{cases}$$

onde a e b são constantes arbitrárias. Note que (10.52) é um sistema *linear* em x e y.

Uma das razões para se utilizar um sistema (10.51) é quando temos de resolver uma EDO não-linear. Suponha

$$y^{(n)} = f(t, y, y', \ldots, y^{(n-1)})$$

com $y^{(i)} = \frac{d^i}{dt^i}y$. Escrevendo

$$x_1 = y, \quad x_2 = y', \quad x_3 = y'', \quad \ldots, \quad x_n = y^{n-1}$$

temos

$$\begin{cases} x_1' = x_2 \\ x_2' = x_3 \\ x_3' = x_4 \\ \ddots \\ x_n' = f(t, x_1, x_2, \ldots, x_n) \end{cases}$$

Tal substituição de variáveis é necessária em muitos casos, de forma a poder utilizar algum "software" que não resolve uma EDO não-linear, porém oferece a solução de sistemas de EDOs. Vejamos alguns exemplos:

Exemplo 10.11 *Obtenha o sistema de EDOs correspondente à equação*

$$\operatorname{sen}(t)y''' + \cos(ty) + \operatorname{sen}(t^2 + y'') + (y')^3 = \log t$$

Solução: *Introduzindo as variáveis $x_1 = y$, $x_2 = y'$ e $x_3 = y''$, temos*

$$\begin{cases} x_1' = x_2 \\ x_2' = x_3 \\ x_3' = (\log t - x_2^3 - \operatorname{sen}(t^2 + x_3 - \cos(tx_1))(\operatorname{sen} t)^{-1} \end{cases}$$

Exemplo 10.12 *Converta o sistema de EDOs não-linear abaixo para um sistema de EDOs lineares:*

$$\begin{cases} (x'')^2 + te^y + y' = x' - x \\ y'y'' - \cos(xy) + \operatorname{sen}(tx'y) = x \end{cases}$$

Solução: *Introduzindo as variáveis $x_1 = x$, $x_2 = x'$, $x_3 = y$ e $x_4 = y''$, temos*

$$\begin{cases} x_1' = x_2 \\ x_2' = \sqrt{x_2 - x_1 - x_4 - te^{x_3}} \\ x_3' = x_4 \\ x_4' = (x_1 - \operatorname{sen}(tx_2x_3) + \cos(x_1x_3))x_4^{-1} \end{cases}$$

Podemos representar de maneira compacta o sistema (10.51) utilizando uma notação matricial. Seja então X um vetor cujas componentes são x_1, x_2, ..., x_n, os quais são funções de t, e F o vetor com componentes f_1, f_2, ..., f_n. Então, um PVI para sistema de EDOs pode ser escrito como

$$\begin{cases} X' = F(t, X) \\ X(t_0) = X_0 \end{cases} \tag{10.53}$$

10.2.8.1 Método da Série de Taylor

O método da série de Taylor, visto na seção 10.2.3, pode ser utilizado nesse caso, devidamente adaptado. Escreve-se a expansão em série para cada variável,

$$x_i(t+h) = x_i(t) + hx_i'(t) + \frac{h^2}{2!}x_i''(t) + O(h^3)$$

e, escrevendo o sistema resultante em forma matricial, vem

$$X(t+h) = X(t) + hX'(t) + \frac{h^2}{2!}X''(t) + O(h^2) \tag{10.54}$$

Note que as derivadas em (10.54) podem necessitar ser calculadas em uma determinada ordem, devido a dependências existentes entre as mesmas, no sistema considerado.

Teoricamente, as equações no sistema (10.54) não necessitam conter t explicitamente. Podemos escrevê-las na forma

$$x_i' = f_i(x_0, x_1, \ldots, x_n)$$

com $x_0 \equiv t$ – cuja equação diferencial correspondente é $x_0' = 1$. O sistema (10.54) pode então ser escrito na *forma autônoma*

$$X' = F(X) \tag{10.55}$$

onde $X = (x_0, x_1, \ldots, x_n)^T$. O exemplo a seguir ilustra essa técnica.

Exemplo 10.13 *Escreva o sistema*

$$\begin{cases} (\text{sen}t)y''' + \cos(ty) + \text{sen}(t^2 + y'') + (y')^3 = \log t \\ y(2) = 7 \\ y'(2) = 3 \\ y''(2) = -4 \end{cases}$$

na forma autônoma.

Solução: *Escrevendo* $x_0 = t$, $x_1 = y$, $x_2 = y'$ *e* $x_3 = y''$, *vem*

$$\begin{cases} x_0' = 1 \\ x_1' = x_2 \\ x_2' = x_3 \\ x_3' = (\log x_0 - x_2^3 - \text{sen}(x_0^2 + x_3) - \cos(x_0 x_1))(\text{sen}x_0)^{-1} \end{cases}$$

com condição inicial $X_0 = (2, 7, 3, -4)^T$.

10.2.8.2 Método de Runge-Kutta

Se um sistema de EDOs em um PVI encontra-se na forma autônoma (10.55), o método de Runge-Kutta de quarta ordem pode ser escrito como

$$\begin{aligned} X(t+h) &= X(t) + \frac{h}{6}(F_1 + 2F_2 + 2F_3 + F_4) \\ F_1 &= hF(X) \\ F_2 &= hF\left(X + \frac{1}{2}F_1\right) \\ F_3 &= hF\left(X + \frac{1}{2}F_2\right) \\ F_4 &= hF(X + F_3) \end{aligned} \tag{10.56}$$

De forma similar, podemos obter variações dos métodos de Runge-Kutta-Fehlberg e de passo múltiplo para um sistema de EDOs na forma autônoma.

10.2.9 Solução via decomposição em autovalores e autovetores

Seja um sistema de equações diferenciais ordinárias lineares com coeficientes constantes, expresso na forma autônoma:

$$\begin{cases} x_1' = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ \ddots \\ x_n' = a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n \end{cases} \tag{10.57}$$

ou,

$$X' = AX \tag{10.58}$$

Se procuramos obter um vetor solução X e tomamos como tal um vetor na forma $X(t) = e^{\lambda t}v$, com $\lambda \in \mathbb{R}$ e v um vetor constante, e substituímos em (10.58), obtemos

$$\lambda e^{\lambda t}v = e^{\lambda t}Av \tag{10.59}$$

e, se

$$Av = \lambda v$$

for satisfeita, então a função vetorial $e^{\lambda t}v$ é solução de (10.58). Agora, para qual λ essa igualdade é satisfeita? Os teoremas a seguir qualificam esse escalar e o vetor v.

Teorema 10.2.10 *Se λ é um autovalor de A e v o autovetor correspondente, então $X(t) = e^{\lambda t}v$ é solução de $X' = AX$.*

Teorema 10.2.11 *Se $A_{n\times n}$ tem um conjunto de autovetores $v_1, v_2, \ldots, v_n$ linearmente independentes, com $Av_i = \lambda_i v_i$, então o espaço solução da equação $X' = AX$ tem uma base $x_i = e^{\lambda_i t}v_i$, para $1 \leq i \leq n$.*

Se A tem a propriedade expressa no teorema 10.2.11, então existe uma matriz não-singular V cujas colunas são os vetores $v_1, v_2, \ldots, v_n$,

$$V_{n\times n} = \begin{bmatrix} v_{11} & v_{12} & \ldots & v_{1n} \\ v_{21} & v_{22} & \ldots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & \ldots & v_{nn} \end{bmatrix}. \tag{10.60}$$

Em forma matricial, podemos escrever $Av_i = \lambda_i v_i$ como

$$AV = V\Lambda \tag{10.61}$$

onde

$$\Lambda_{n\times n} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}. \tag{10.62}$$

Agora, a resolução da equação (10.58) pode ser bastante simplificada se fizermos a troca de variáveis $X = VY$. Como V é não-singular, podemos escrever

$$Y' = V^{-1}X' = V^{-1}AX = V^{-1}AVY = \Lambda Y$$

a qual é uma equação muito mais simples de se resolver, dada a forma de Λ. As equações em $Y' = \Lambda Y$ são ditas *desacopladas* e podem ser resolvidas separadamente, como mostra o exemplo a seguir.

Exemplo 10.14 *Resolva o PVI* $X' = AX$ *com*

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad X(0) = \begin{bmatrix} 5 \\ 7 \\ 6 \end{bmatrix}$$

Solução: *Os autovalores de* A *são* $\lambda_1 = 1$, $\lambda_2 = 0$ *e* $\lambda_3 = -1$*; seus autovetores correspondentes são* $v_1 = (1,0,0)^T$, $v_2 = (0,1,0)^T$ *e* $v_3 = (1,0,-2)^T$. *Logo,*

$$V = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \quad V^{-1} = \begin{bmatrix} 1 & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{2} \end{bmatrix}$$

Se $Y = (y_1, y_2, y_3)^T$, *então* $Y' = \Lambda Y$, *com*

$$\Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} = V^{-1}AV$$

de onde podemos escrever:

$$\begin{cases} y_1' = y_1 \\ y_2' = 0 \\ y_3' = -y_3 \end{cases}, \quad Y(0) = V^{-1}X(0) = \begin{bmatrix} 8 \\ 7 \\ -3 \end{bmatrix}$$

cuja solução é

$$y_1 = 8e^t, \quad y_2 = 7, \quad y_3 = -3e^{-t}.$$

Como $X = VY$, *a solução é, por fim,*

$$x_1 = 8e^t - 3e^{-t}, \quad x_2 = 7, \quad x_3 = 6e^{-t}$$

10.2.9.1 O expoente de uma matriz

A solução da equação (10.58) pode ser expressa através da forma e^A, a qual é definida como

$$e^A = I + A + \frac{1}{2!}A^2 + \frac{1}{!}A^3 + \ldots \tag{10.63}$$

e, para $t \in \mathbb{R}$, então $tA = At$, de onde

$$e^{At} = \sum_{k=0}^{\infty} \frac{t^k}{k!}A^k \tag{10.64}$$

e, diferenciando em relação a t, temos

$$\frac{d}{dt}e^{At} = Ae^{At} \tag{10.65}$$

Assim, a solução do problema (10.5) é

$$X(t) = e^{At}X(0) \tag{10.66}$$

Se a matriz A for diagonalizável, i.e., $AV = V\Lambda$, com V e Λ definidos conforme (10.60) e (10.62), então a solução de (10.5) pode ser escrita como

$$X = VY = V\left(e^{\Lambda t}V^{-1}X(0)\right) = V\begin{bmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ & & & e^{\lambda_n t} \end{bmatrix} V^{-1}X(0) \tag{10.67}$$

Em caso contrário, uma análise através da forma de Jordan da matriz A deve ser realizada.

10.2.10 Equações rígidas

A rigidez em um sistema de equações diferenciais refere-se a uma grande disparidade nas escalas de tempo dos componentes do vetor solução. Como conseqüência, métodos numéricos que são satisfatórios para outros sistemas, normalmente falham em sistemas rígidos; isto acontece quando a estabilidade no processo numérico ocorre apenas quando passos de integração muito pequenos podem ser empregados. Tais sistemas surgem em diferentes áreas de aplicação:

Controle de naves espaciais: a trajetória de vôo para re-entrada na atmosfera deve ser suave, mas rápidas correções devem ser feitas caso ocorrer qualquer desvio;

Monitoração de processos químicos: quaisquer mudanças de natureza física e química podem ter uma grande variação nas escalas de tempo envolvidas;

Circuitos eletrônicos: transientes da ordem de microssegundos são impostos ao circuito como um todo.

Como exemplo, vejamos o comportamento do método de Euler (10.13) para o problema

$$\begin{cases} x' = \lambda x \\ x(0) = 1 \end{cases} \tag{10.68}$$

o qual pode ser escrito, nesse caso, como

$$x_{n+1} = x_n + h\lambda x_n = (1 + h\lambda)x_n, \quad x_0 = 1 \tag{10.69}$$

Então, na n-ésima iteração,

$$x_n = (1 + h\lambda)^n \tag{10.70}$$

mas a solução de (10.68) é

$$x(t) = e^{\lambda t}$$

a qual tende a zero, se $\lambda < 0$, quando t tende a infinito. Ora, a equação (10.70) só tende a zero se e somente se $|1 + h\lambda| < 1$. Somos, então, obrigados a escolher h tal que $1 + h\lambda > -1$; como $\lambda < 0$, temos que $h < -2/\lambda$.

Por exemplo, se $\lambda = -20$, $h < 0,1$, apesar da solução que queremos obter ser praticamente plana (e quase zero) imediatamente após $t = 0$, quando $x = 1$ (note que $x(t) = e^{-20t} \leq 2,1 \times 10^{-9} \mid t \geq 1$). O método de Euler, então, procederá com passos pequenos, quando o problema indica que passos grandes devem ser tomados – isso é um aspecto que caracteriza a rigidez do problema. A função e^{-20t} é dita transiente porque seu efeito físico é de pouca duração (pois decai rapidamente para zero). Desejamos então um procedimento numérico que permita acompanhar funções transientes com passos pequenos até que o efeito transiente seja desprezível, quando então passos grandes podem ser tomados.

Já o *método implícito de Euler*, definido por

$$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1}), \quad n \geq 0 \tag{10.71}$$

apresentará, para esse problema, uma restrição que é satisfeita para quaisquer valores de $h > 0$. Para o problema em questão, o método implícito é escrito como

$$x_{n+1} = x_h + h\lambda x_{n+1}, \quad x_0 = 1$$

ou

$$x_{n+1} = (1 - h\lambda)^{-1} x_n$$

de onde, na n-ésima iteração,

$$x_n = (1 - h\lambda)^{-n}$$

e, para $\lambda < 0$, é necessário satisfazer $|1 - h\lambda|^{-1} < 1$, o que é verdadeiro para qualquer valor positivo de h.

10.3 Problemas de Valor de Fronteira

Um *problema de valor de fronteira* (PVF) caracteriza-se pela especificação de valores para a função $x(t)$ nos extremos do intervalo de integração em t,

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha,\ x(b) = \beta \end{cases} \tag{10.72}$$

Esse tipo de problema é mais difícil de ser resolvido do que um PVI, conforme veremos a seguir.

Para um PVI, havíamos assumido que, se a função $x(t)$ fosse "suave", então provavelmente (sujeito também a outras condições), o problema teria solução. Em um PVF, no entanto, isto não se aplica, como pode-se ver no exemplo a seguir.

Exemplo 10.15 *Considere o PVF*

$$\begin{cases} x'' = x' \\ x(0) = 3,\ x(\pi) = 7 \end{cases} \tag{10.73}$$

cuja solução é $x(t) = A\text{sen}t + B\cos t$. *Usando essa expressão e igualando aos valores especificados para* x, *vem*

$$\begin{cases} 3 = x(0) = A\text{sen}0 + B\cos 0 = B \\ 7 = x(\pi) = A\text{sen}\pi + B\cos\pi = -B \end{cases}$$

o que é uma contradição e, logo, o PVF (10.73) não tem solução, apesar de f *ser uma função "suave".*

O teorema a seguir, por Keller (1968), fala da existência de solução de um PVF escrito numa forma bastante particular.

Teorema 10.3.1 *O problema de valor de fronteira*

$$\begin{cases} x'' = f(t, x) \\ x(0) = 0,\ x(1) = 0 \end{cases}$$

tem solução única se $\frac{\partial f}{\partial x}$ *é contínua, não-negativa, e limitada na tira* $0 \le t \le 1$, $-\infty < x < \infty$.

Considere então o exemplo a seguir:

Exemplo 10.16 *O PVF*

$$\begin{cases} x'' = (5x + \text{sen}(3x))e^t \\ x(0) = x(1) = 0 \end{cases}$$

tem solução única, pois $\frac{\partial f}{\partial x} = (5 + 3\cos(3x))e^t$, *a qual é contínua em* $0 \le t \le 1$, $-\infty < x < \infty$. *Além disso, é limitada por* $8e$ *e não assume valores negativos, pois* $3\cos(3x) \ge -3$.

O teorema 10.3.1 apresenta o PVF em uma forma bastante particular. A fim de podermos utilizá-lo para problemas mais gerais, é necessário fazer uma troca de variáveis. Considere, então, o PVF

$$\begin{cases} x'' = f(t, x) \\ x(a) = \alpha,\ x(b) = \beta \end{cases} \tag{10.74}$$

com $x = x(t)$. Escrevendo $t = a + (b - a)s$, reduzimos (10.74) à forma requerida pelo teorema 10.3.1 (note que $s = 0$ corresponde a $t = a$ e $s = 1$ a $t = b$). Escrevendo, agora,

$$y(s) = x(a + \lambda s)$$

com $\lambda = b - a$, obtemos

$$\begin{aligned} y' &= \lambda x'(a + \lambda s) \\ y'' &= \lambda^2 x''(a + \lambda s) \end{aligned}$$

e $y(0) = x(a) = \alpha$, $y(1) = x(b) = \beta$. Então, se x é solução de (10.74), y é solução de

$$\begin{cases} y''(s) = \lambda^2 f(a + \lambda s, y(s)) \\ y(0) = \alpha,\ y(1) = \beta \end{cases} \tag{10.75}$$

e, se y é solução de (10.75),

$$x(t) = y\left(\frac{t-a}{b-a}\right)$$

é solução de (10.74), conforme o teorema a seguir.

Teorema 10.3.2 *Considere os seguintes PVF:*

$$\begin{cases} x'' = f(t, x) \\ x(a) = \alpha,\ x(b) = \beta \end{cases} \tag{10.76}$$

$$\begin{cases} y'' = g(t, y) \\ y(0) = \alpha,\ y(1) = \beta \end{cases} \tag{10.77}$$

onde $g(p, q) = (b-a)^2 f(a + (b-a)p, q)$. *Então, se* y *é solução de (10.77),*

$$x(t) = y\left(\frac{t-a}{b-a}\right)$$

é solução de (10.76); e, se x *é solução de (10.76), então*

$$y(a + (b-a)t)$$

é solução de (10.77).

Prova:

$$\begin{aligned} x(a) &= y\left(\frac{a-a}{b-a}\right) = y(0) = \alpha \\ x(b) &= y\left(\frac{b-a}{b-a}\right) = y(a) = \beta \\ x'(t) &= y'\left(\frac{t-a}{b-a}\right)\frac{1}{b-a} \\ x''(t) &= y''\left(\frac{t-a}{b-a}\right)\frac{1}{(b-a)^2} = g\left(\frac{t-a}{b-a}, y\left(\frac{t-a}{b-a}\right)\right)\frac{1}{(b-a)^2} \\ &= (b-a)^2 f\left(a + (b-a)\frac{t-a}{b-a}, y\left(\frac{t-a}{b-a}\right)\right)\frac{1}{(b-a)^2} \\ &= f(t, x(t)) \diamond \end{aligned}$$

A seguir, veremos alguns dos métodos disponíveis para resolver um PVF.

10.3.1 Método do disparo

Considere o problema (10.72). Uma maneira de resolvê-lo é através da resolução do problema (relacionado) de valor inicial

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha,\ x'(a) = z \end{cases} \tag{10.78}$$

e integrar a equação no intervalo $a \le t \le b$, a fim de obter uma solução aproximada, na esperança de que $x(b) = \beta$. Se tal não ocorrer, a estimativa para $x'(a)$ pode ser modificada, e o processo de integração repetido novamente.

A solução de (10.78) é x_z, onde o subscrito em z indica a derivada. A idéia, aqui, é relacionar z tal que $x_z(b) = \beta$. Escrevendo

$$\phi(z) = x_z(b) - \beta \tag{10.79}$$

podemos reduzir o problema de resolver (10.72) a encontrar a raiz da função não-liner $\phi(z)$, para o qual métodos como o da bissecção, secante e de Newton (vide Capítulo 2) podem ser utilizados. Note, no entanto, que a função ϕ é custosa de se avaliar, pois envolve a solução de um PVI! Por isso, deve-se procurar minimizar seu impacto no método como um todo, por exemplo, utilizando passos de integração pequenos apenas quando $\phi(z)$ é próximo de zero.

Para problemas *lineares*, o método da secante obtém a solução exata em uma única iteração. Se o PVF tiver a forma

$$\begin{cases} x'' = u(t) + v(t)x + w(t)x' \\ x(a) = \alpha,\ x(b) = \beta \end{cases} \tag{10.80}$$

com u, v e w funcões contínuas em $a \leq x \leq b$. Suponha que (10.80) foi resolvida para duas condições iniciais diferentes, obtendo soluções x_1 e x_2,

$$\begin{cases} x_1(a) = \alpha,\ x_1'(a) = z_1 \\ x_2(a) = \alpha,\ x_2'(a) = z_2 \end{cases} \tag{10.81}$$

Combinando linearmente x_1 e x_2, temos

$$y(t) = \lambda x_1(t) + (1 - \lambda)x_2(t) \tag{10.82}$$

com λ um parâmetro. Note que $y(a) = \alpha$, satisfazendo uma das condições de (10.81) (independente do valor de λ). Para a outra condição, selecionamos λ tal que $y(b) = \beta$, i.e.

$$\beta = y(b) = \lambda x_1(b) + (1 - \lambda)x_2(b)$$

$$\lambda = \frac{\beta - x_2(b)}{x_1(b) - x_2(b)} \tag{10.83}$$

Assim, podemos obter ambas as soluções ao mesmo tempo, resolvendo dois PVI simultaneamente:

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha,\ x'(a) = 0 \end{cases} \qquad \begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha,\ x'(a) = 1 \end{cases}$$

onde $f(t, x, x') = u(t) + v(t)x + w(t)x'$, cujas soluções são x_1 e x_2, respectivamente. Procedemos então à formulação de um sistema de EDOs na forma autônoma,

$$\begin{cases} x_0 = 1 \\ x_1' = x_3 \\ x_2' = x_4 \\ x_3' = f(x_0, x_1, x_3) \\ x_4' = f(x_0, x_2, x_4) \end{cases} \tag{10.84}$$

e, para resolvermos (10.80), executamos os seguintes passos:

1. Resolver (10.84), com os valores discretos de $x_1(t_i)$ e $x_2(t_i)$ para $a \leq t_0 \leq t_i \leq t_m = b$, os quais devem ser armazenados em vetores;
2. Calcular o valor de λ por (10.83);
3. Calcular $y(t_i)$ por (10.82), para cada t_i.

O teorema a seguir enuncia a solução do PVF linear.

Teorema 10.3.3 *Se o PVF linear tem solução, então x_1 é uma solução, ou $x_1(b) - x_2(b) \neq 0$ e y é uma solução.*

10.3.2 Método de Newton

O método de Newton (vide seção 2.4) pode ser usado para resolver o PVF não-linear. Seja x_z a solução do problema

$$\begin{cases} x_z'' = f(t, x_z, x_z') \\ x_z(a) = \alpha,\ x_z'(a) = z \end{cases} \tag{10.85}$$

e z é tal que $\phi(z) = x_z(b) - \beta = 0$. Relembrando, a equação governante do método de Newton para a função ϕ é

$$z_{n+1} = z_n - \frac{\phi(z_n)}{\phi'(z_n)} \tag{10.86}$$

A derivada ϕ' é determinada diferenciando parcialmente com respeito a z as funções componentes em (10.85):

$$\begin{cases} \frac{\partial x_z}{\partial z} = \frac{\partial f}{\partial t}\frac{\partial t}{\partial z} + \frac{\partial f}{\partial x_z}\frac{\partial x_z}{\partial z} + \frac{\partial f}{\partial x_z'}\frac{\partial x_z'}{\partial z} \\ \frac{\partial}{\partial z}x_z(a) = 0,\ \frac{\partial}{\partial z}x_z'(a) = 1 \end{cases} \tag{10.87}$$

Introduzindo a variável $v = \frac{\partial x_z}{\partial z}$ e simplificando, vem

$$\begin{cases} v'' = f_{x_z}(t, x_z, x_z')v + fx_z'(t, x_z, x_z')v' \\ v(a) = 0,\ v'(a) = 1 \end{cases} \tag{10.88}$$

a qual é denominada de *primeira equação variacional*. A equação (10.88) pode ser resolvida, a cada iteração, juntamente com a equação (10.85). Ao final, $v(b)$ será obtida, i.e.

$$v(b) = \frac{\partial x_z(b)}{\partial z} = \phi'(z)$$

10.3.3 Método da colocação

O *método da colocação* é aplicável a muitos problemas. Suponha que é dado um operador linear $\mathcal{L}$ (integral ou diferencial) e desejamos resolver a equação

$$\mathcal{L}u = w \tag{10.89}$$

com w conhecido. Procuramos, então, resolvê-la selecionando um conjunto de vetores $V = \{v_1, v_2, \ldots, v_n\}$ e, combinando-os linearmente, obter a solução u, na forma

$$u = c_1v_1 + c_2v_2 + \ldots + c_nv_n \tag{10.90}$$

Como $\mathcal{L}$ é um operador linear, podemos escrever

$$\mathcal{L}u = \sum_{j=1}^{n} c_j \mathcal{L}v_j$$

logo,

$$\sum_{j=1}^{n} c_j \mathcal{L}v_j = w \tag{10.91}$$

De forma geral, usualmente não podemos resolver (10.91) e obter os coeficientes c_i; porém, podemos exigir que os dois termos em (10.91) sejam idênticos em determinados pontos. No *método da colocação*, os vetores u, w e v_j são funções no mesmo domínio, e temos

$$\sum_{j=1}^{n} c_j(\mathcal{L}v_j)(t_i) = w(t_i), \quad 1 \leq i \leq n \tag{10.92}$$

o qual reduz-se a um sistema de n equações lineares. As funções v_j e os pontos t_i devem ser escolhidos tal que a matriz cujas entradas são $(\mathcal{L}v_j)(t_i)$ seja *não-singular*.

Podemos usar diferentes funções para calcular os vetores v_j, porém as chamadas "B-splines" são bastante adequadas. Suponha um problema na forma

$$\begin{cases} u'' + pu' + qu = w \\ u(a) = \alpha,\ u(b) = \beta \end{cases} \tag{10.93}$$

onde $\mathcal{L}u = u'' + pu' + qu$. Como necessitamos de funções com as primeira e segunda derivadas contínuas, vamos considerar aqui as "B-splines" B_i^k cúbicas, apesar de nada impedir que se usem aquelas de grau k maior. Assumimos, também, que os nós t_i da "B-spline" são igualmente espaçados: $t_{i+1} - t_i = h$, e os nós serão os pontos de colocação.

Seja n o número de funções a serem usadas e, portanto, o número de coeficientes a serem determinados; logo, necessitamos de n condições para determiná-los. O problema (10.93) apresenta duas condições de fronteira, as quais devem satisfazer

$$\sum_{j=1}^{n} c_j(\mathcal{L}v_j)(a) = \alpha, \quad \sum_{j=1}^{n} c_j(\mathcal{L}v_j)(b) = \beta \tag{10.94}$$

Para as demais $n-2$ condições, temos

$$\sum_{j=1}^{n} c_j(\mathcal{L}v_j)(t_i) = w(t_i), \quad 1 \le i \le n-2 \tag{10.95}$$

e podemos, então, definir

$$h = \frac{b-a}{n-3} \tag{10.96}$$

$$t_i = a + (i-1)h, \quad i = 0, \pm 1, \pm 2, \ldots \tag{10.97}$$

Os nós t_i pertencentes ao intervalo $[a, b]$ são

$$a = t_1 < t_2 < \ldots < t_{n-3} < t_{n-2} = b$$

os quais são os *pontos de colocação*. Para definirmos as "B-splines" B_j^3, necessitamos de alguns pontos fora do intervalo $[a, b]$ os quais, por (10.97), encontram-se dispostos assim:

$$t_{-3} < t_{-2} < t_{-1} < t_0 < \underset{t_1}{a} < t_2 < t_3 < \ldots < t_{n-4} < t_{n-3} < \underset{t_{n-2}}{b} < t_{n-2} < t_{n-2} < t_{n-1} < t_n < t_{n+1}$$

Ora, as "B-splines" cúbicas tem a forma segundo a figura 10.2, e estamos interessados naquelas

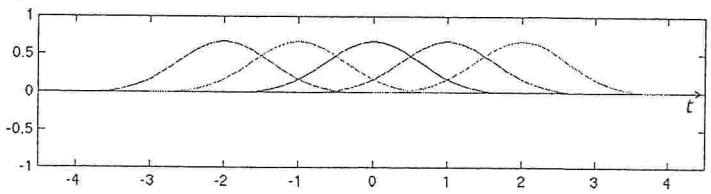

Figura 10.2: *Forma das "B-splines" cúbicas.*

que não são identicamente nulas no intervalo $[a, b]$, as quais são as "B-splines" B_{-2}^3, B_{-1}^3, B_0^3, B_1^3, $\ldots$, B_{n-3}^3. Assim, podemos escrever

$$v_j = B_{j-3}^3, \quad 1 \le j \le n.$$

Como utilizamos espaçamento igual entre os nós, podemos defini-las através de uma única formulação, $B^3(t)$, dada por

$$B^3(t) = \begin{cases} \frac{(t+2)^3}{6}, & -2 \le t \le -1 \\ \frac{1+3(t+1)+3(t+1)^2-3(t+1)^3}{6}, & -1 \le t \le 0 \\ \frac{1+3(1-t)+3(1-t)^2-3(1-t)^3}{6}, & 0 \le t \le 1 \\ \frac{(2-t)^3}{6}, & 1 \le t \le 2 \\ 0, & c.c. \end{cases} \tag{10.98}$$

a qual tem a forma

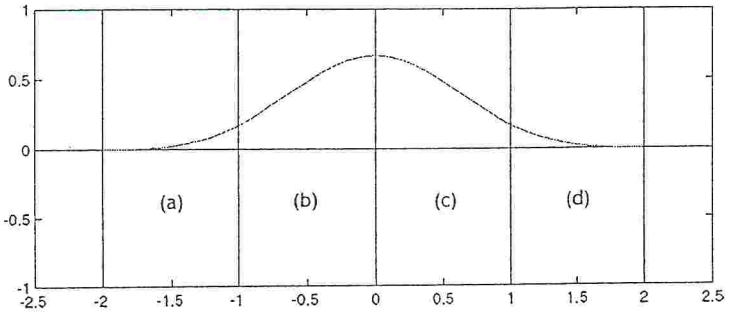

Figura 10.3: *"B-spline" cúbica: (a)* $(t+2)^3)/6$, *(b)* $(1+3(t+1)+3(t+1)^2-3(t+1)^3)/6$, *(c)* $(1+3(1-t)+3(1-t)^2-3(1-t)^3)/6$, *(d)* $((2-t)^3)/6$.

O exemplo a seguir mostra como obter os nós.

Exemplo 10.17 *Suponha que desejamos obter m pontos de colocação, com* $m = 4$. *Como dois pontos extras são dados pelas condições de fronteira, teremos quatro pontos internos e, como cada "B-spline" cúbica necessita de cinco pontos, teremos de obter dois pontos a mais em cada extremo do intervalo* $[a, b]$. *Como* $m = 6$, *teremos* $n = 8$ *pontos, de onde* $h = 1/5$ *e os nós serão:*

i	-1	0	1	2	3	4	5	6	7	8
t_i	$-2/5$	$-1/5$	0	$1/5$	$2/5$	$3/5$	$4/5$	1	$6/5$	$7/5$

10.3.4 Derivação numérica

Um problema num domínio contínuo pode ser discretizado de forma que as variáveis dependentes sejam consideradas existentes apenas para pontos discretos. Desta maneira, as derivadas são aproximadas por diferenças. O método de diferenças finitas é baseado em algumas propriedades da série de Taylor e em aplicações diretas da definição de derivadas. Ele é o mais antigo dos métodos aplicados na obtenção de soluções numéricas de equações diferenciais. A idéia deste método de aproximação é bastante simples.

Como exemplo, toma-se a derivada de uma função $f(x)$ no ponto x, que é definida por

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}. \tag{10.99}$$

Se h é suficientemente pequeno, a expressão do lado direito é uma aproximação para o valor exato de $f'(x)$. Esta aproximação pode ser melhorada com a redução de h. Entretanto, para qualquer valor finito de h, um erro, que tende a zero para h tendendo a zero, é introduzido. Este é o chamado *erro de truncamento*. A potência de h com a qual ele tende a zero é chamada de *ordem da aproximação a diferenças* e pode ser obtida através de um desenvolvimento em série de Taylor de $f(x+h)$ em torno do ponto x. Desenvolvendo $f(x+h)$, obtém-se

$$f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2} f''(x) + \ldots \tag{10.100}$$

e portanto,

$$\frac{f(x+h)-f(x)}{h} = f'(x) + \frac{h}{2} f''(x) + \ldots. \tag{10.101}$$

Diz-se que essa aproximação para $f'(x)$ é de primeira ordem em h e escreve-se

$$f'(x) = \frac{f(x+h)-f(x)}{h} + O(h), \tag{10.102}$$

indicando que o erro de truncamento é de $O(h)$, isto é, tende a zero quando h tende a zero.

Para entender como estas fórmulas são aplicadas na aproximação das derivadas considera-se uma discretização do eixo x. O domínio contínuo é substituído por um conjunto discreto de $n+1$ pontos x_i, $i = 0, 1, \ldots, n$, com espaçamento constante e igual a h entre os pontos, conforme representação na figura 10.4. Denota-se por f_i os valores da função $f(x)$ nos pontos $x_i = ih$ (ou seja, f_i é igual a $f(x_i)$).

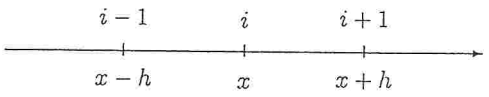

Figura 10.4: *Representação esquemática dos pontos no eixo das abscissas.*

As seguintes aproximações em diferenças finitas podem ser definidas para a primeira derivada no ponto $x = x_i$, $f'(x)_i$:

$$f'_i = \frac{f_{i+1}-f_i}{h} + O(h), \tag{10.103}$$

$$f'_i = \frac{f_i - f_{i-1}}{h} + O(h). \tag{10.104}$$

A primeira fórmula é denominada *diferença ascendente* e a segunda, *diferença descendente*. Ambas são aproximações de primeira ordem para $f'(x)_i$. Outras fórmulas, com diferentes ordens de aproximação, podem ser obtidas. A mais comum é a de segunda ordem, obtida conforme descrição abaixo.

Fazendo duas expansões diferentes em série de Taylor para a primeira derivada,

$$f(x+h) = f(x) + h\, f_x(x) + \frac{h^2}{2} f''(x)(x) + \ldots \tag{10.105}$$

e

$$f(x-h) = f(x) - h\, f_x(x) + \frac{h^2}{2} f''(x) + \ldots, \tag{10.106}$$

e subtraindo (10.106) de (10.105), obtém-se

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + O(h^2), \tag{10.107}$$

que é uma aproximação de segunda ordem para $f'(x)$. Na notação de diferenças finitas, esta expressão fica

$$f'_i = \frac{f_{i+1}-f_{i-1}}{2h} + O(h^2), \tag{10.108}$$

cuja ordem de precisão é maior do que em (10.103) e (10.104).

Uma ilustração gráfica para os três tipos de aproximações para a derivada de primeira ordem, dados por (10.103), (10.104) e (10.108), é fornecida pela figura (10.5).

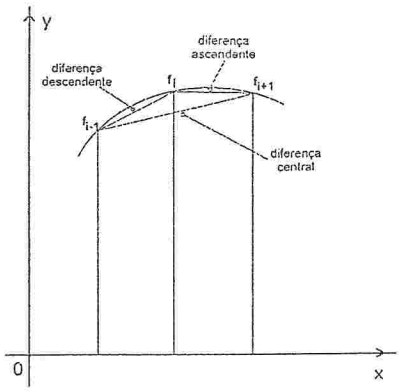

Figura 10.5: Ilustração gráfica para as derivadas de primeira ordem [8].

Ainda, para exemplificar outras fórmulas de diferenças finitas, são apresentadas as aproximações centrais de segunda ordem para as derivadas abaixo:

$$f_i^{(2)} = \frac{f_{i+1} - 2\,f_i + f_{i-1}}{h^2} + O(h^2), \tag{10.109}$$

$$f_i^{(3)} = \frac{f_{i+2} - 2\,f_{i+1} + 2\,f_{i-1} - fi - 2}{2\,h^3} + O(h^2), \tag{10.110}$$

$$f_i^{(4)} = \frac{f_{i+2} - 4\,f_{i+1} + 6\,f_i - 4\,f_{i-1} + f_{i-2}}{h^4} + O(h^2), \tag{10.111}$$

Na verdade, fórmulas em diferenças para a aproximação de derivadas podem ser construídas com um número arbitrário de pontos adjacentes. Na prática, em qualquer esquema numérico, é preciso fazer o balanço entre a ordem de precisão e o número de pontos simultaneamente envolvidos nos cálculos.

Exemplo 10.18 *A partir dos valores abaixo, calcular* $f'(1,4)$ *usando diferenças ascendentes, descendentes e centrais.*

x	1,2	1,3	1,4	1,5	1,6
$f(x)$	1,5095	1,6984	1,9043	2,1293	2,3756

1. *Diferenças ascendentes:*

$$f'(1,4) = \frac{f(1,5) - f(1,4)}{0,1} = 2,2500$$

2. *Diferenças descendentes:*

$$f'(1,4) = \frac{f(1,4) - f(1,3)}{0,1} = 2,059$$

3. *Diferenças centrais:*

$$f'(1,4) = \frac{f(1,5) - f(1,3)}{2 \cdot 0,1} = 2,1545$$

Exemplo 10.19 *Seja* $f(x) = \cos x$. *Calcule aproximações para* $f''(0,8)$ *com* $h = 0,1$, $h = 0,01$, $h = 0,001$. *Utilize* 9 *casas decimais em seus cálculos. Depois, compare com o valor real* $f''(0,8) = -\cos 0,8$.

10.3.5 Solução por diferenças-finitas

Considere novamente o problema (10.72). Podemos resolvê-lo se substituirmos as derivadas por aproximações em diferenças-finitas, conforme visto anteriormente.

Para tanto, particionamos o intervalo $[a, b]$ nos pontos t_i,

$$a = t_0 < t_1 < \ldots < t_n < t_{n+1} = b$$

espaçados igualmente entre si, i.e.

$$t_i = a + ih, \quad h = \frac{b-a}{n+1}, \quad 0 \leq i \leq n+1$$

Denotando o valor aproximado $x(t_i)$ por y_i, podemos reescrever o problema (10.72) na *forma discreta*

$$\begin{cases} y_0 = \alpha \\ \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} = f\left(t_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}\right), \ 1 \leq i \leq n \\ y_{n+1} = \beta \end{cases} \tag{10.112}$$

onde as derivadas y' e y'' são aproximadas por aproximações *centrais*, conforme as equações (10.108) e (10.109).

De forma geral, o problema (10.112) reduz-se a um *sistema não-linear de equações*, como pode ser visto no exemplo a seguir.

Exemplo 10.20 *Suponha $f(t, x, x') = x^t + 2x'$. Usando (10.112), para $n = 2$, temos:*

$$\begin{cases} y_0 = \alpha \\ y_0 - 2y_1 + y_2 = h^2 \left(y_1^{t_1} + 2\left(\frac{y_2 - y_0}{2h}\right)\right) \\ y_1 - 2y_2 + y_3 = h^2 \left(y_2^{t_2} + 2\left(\frac{y_3 - y_1}{2h}\right)\right) \\ y_3 = \beta \end{cases}$$

e, procedendo às substituições possíveis, chegamos ao seguinte sistema não-linear de equações:

$$\begin{bmatrix} 1 & & & \\ & -(2+h^2) & (1-h) & \\ & (1+h) & -(2+h^2) & \\ & & & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1^{t_1} \\ y_2^{t_2} \\ y_3 \end{bmatrix} = \begin{bmatrix} \alpha \\ (-(1+h)\alpha \\ (h-1)\beta \\ \beta \end{bmatrix}$$

o qual deve ser resolvido através de um método específico como, por exemplo, o método de Newton.

10.3.5.1 O caso linear

Se o PVF é da forma (10.80), então o sistema (10.112) reduzir-se-á a um conjunto de equações *lineares*, as quais podem ser escritas como

$$\begin{cases} y_0 = \alpha \\ a_i y_{i-1} + d_i y_i + c_i y_{i+1} = b_i \ 1 \leq i \leq n \\ y_{n+1} = \beta \end{cases} \tag{10.113}$$

com

$$\begin{aligned} u_i &= u(t_i) \\ v_i &= v(t_i) \\ w_i &= w(t_i) \\ a_i &= -1 - \frac{h}{2} w_{i+1} \\ d_i &= 2 + h^2 v_i \\ c_i &= -1 + \frac{h}{2} w_i \\ b_i &= -h^2 u_i \end{aligned}$$

o que nos permite escrever o sistema de equações lineares como

$$\begin{bmatrix} d_1 & c_1 & & & & \\ a_1 & d_2 & c_2 & & & \\ & a_2 & d_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-2} & d_{n-1} & c_{n-1} \\ & & & & a_{n-1} & d_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 - a_0\alpha \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n - c_n\beta \end{bmatrix} \tag{10.114}$$

o qual apresenta as seguintes características:

1. É tridiagonal;
2. Se h é pequeno e $v_i > 0$, então (10.114) é *diagonal dominante*, pois

$$|\, 2 + h^2 v_i \,| > \left| 1 + \frac{h}{2} w_i \right| + \left| 1 - \frac{h}{2} w_i \right| = 2,$$

   assumindo que $|\, hw_i/2 \,| \leq 1$.

3. $$|\, d_i \,| - |\, c_i \,| - |\, a_{i-1} \,| = 2 + h^2 v_i - \left(1 - \frac{h}{2} w_i\right) - \left(1 + \frac{h}{2} w_i\right) = h^2 v_i$$

10.4 Exercícios

Exercício 10.1 *O núcleo radioativo do plutônio decai de acordo com a equação diferencial*

$$\frac{dQ}{dt} = -0,0525\, Q$$

Se $50\,mg$ *de plutônio* 241 *estiverem presentes numa amostra no dia de hoje, determine quanto plutônio existirá daqui a* 2 *anos. Considere* $h = 1$ *e* $h = 0,5$. *Discuta seus resultados.*

Exercício 10.2 *Um capital de* $R\$1000,00$ *é depositado em uma conta. Sabendo que sobre ele incide uma taxa de* 10% *de juros compostos ao ano, determine qual será o saldo na conta ao final de* 5 *anos.*

Exercício 10.3 *Em psicologia, a lei de Weber-Fechner para a resposta-estímulo diz que a taxa de variação* $\frac{dR}{dS}$ *da reação* R *é inversamente proporcional ao estímulo. O valor inicial é o nível mínimo de estímulo que pode ser consistentemente detectado. O problema de valor inicial para este modelo é*

$$\begin{aligned} R' &= \frac{k}{S} \\ R(s_0) &= 0 \end{aligned}$$

Suponha que $s_0 = 0,1$ *e use o método de Heun com* $h = 0,1$ *para resolver*

$$\begin{aligned} R' &= \frac{1}{S} \\ R(0,1) &= 0 \end{aligned}$$

no intervalo $[0,1; 0,5]$.

Exercício 10.4 *Usar o método de Taylor de 4^a ordem para resolver*

$$\begin{cases} x' &= \cos t - \operatorname{sen} x + t^2 \\ x(-1) &= 3 \end{cases}$$

com $h = 10^{-2}$, $t_0 = -1$, $t_1 = 1$ e $x_0 = 3$. Após, usar a solução obtida como um novo valor de x_0, e repetir o processo, desta vez com $x_0 = -10^{-2}$. Explique o que ocorre.

Exercício 10.5 *Um paraquedista salta de um avião e até o momento que ele abre o pára-quedas, a resistência do ar é proporcional a $v^{\frac{3}{2}}$. Assuma que o intervalo de tempo é $[0, 2]$ e que a equação diferencial para a direção vertical é*

$$\begin{aligned} v' &= 32 - 0,032\, v^{\frac{3}{2}} \\ v(0) &= 0 \end{aligned}$$

Use o método de Taylor de segunda ordem com $h = 0,5$ e encontre a solução para este problema.

Exercício 10.6 *Assuma que a curva $P(t)$ para uma determinada população obedeça a equação diferencial para uma curva logística $P' = a\,P - b\,P^2$. Seja t o tempo em anos e $h = 10$ o passo. Os valores $a = 0,02$ e $b = 0,00004$ produzem um modelo para a população. Considerando que no ano de 1990 a população era $76,1$ milhões, obtenha, usando o método de Heun, uma estimativa para esta população no ano de 2010.*

Exercício 10.7 *Resolva o exercício 10.4 usando o método de Runge-Kutta de 2^a ordem.*

Exercício 10.8 *Resolva o exercício 10.4 usando o método de Runge-Kutta de 4^a ordem.*

Exercício 10.9 *Use o método de Runge-Kutta 4^a ordem para resolver*

$$\begin{cases} x' &= e^{xt} + \cos(x - t) \\ x(1) &= 3 \end{cases}$$

com $h = 0,01$. Apresente o último valor para x antes de ocorrer "overflow".

Exercício 10.10 *Supondo o método de Runge-Kutta de 3^a ordem,*

$$\begin{aligned} x(t+h) &= x(t) + \frac{1}{9}(2F_1 + 3F_2 + 4F_3) \\ F_1 &= hf(t, x) \\ F_2 &= hf\left(t + \frac{1}{2}h, x + \frac{1}{2}F_1\right) \\ F_3 &= hf\left(t + \frac{3}{4}h, x + \frac{3}{4}F_2\right) \end{aligned}$$

mostre que, para $x' = x + t$, ele é equivalente ao método de Taylor de 3^a ordem.

Exercício 10.11 *Em uma reação química, uma molécula de A se combina com uma molécula de B para formar uma molécula do produto químico C. Sabe-se que a concentração, $y(t)$, no tempo t, é a solução do problema de valor inicial:*

$$\begin{aligned} y' &= k\,(a - y)\,(b - y) \\ y(0) &= 0 \end{aligned}$$

onde k é uma constante positiva e a e b são as concentrações iniciais de A e de B, respectivamente. Suponha que $k = 0,01$, $a = 70$ milimoles/litro e $b = 50$ milimoles/litro. Use os métodos de Runge-Kutta de ordem $N = 2$ e $N = 4$ com $h = 0,5$ para encontrar a solução em $[0, 2]$.

Exercício 10.12 *Resolva o PVI*

$$\begin{cases} x' &= x^2 \\ x(0) &= 1 \end{cases}$$

no intervalo $0 \leq t \leq 2$ *usando o método de Runge-Kutta-Fehlberg. Compare com a solução analítica dada por* $x(t) = (1-t)^{-1}$. *Explique o que ocorre* perto *da discontinuidade em* $t = 1$, *quando se usa o algoritmo 10.2.3.*

Exercício 10.13 *Determine as características numéricas do método de passo múltiplo cuja equação é*

$$x_n + 4x_{n-1} - 5x_{n-2} = h(4f_{n-1} + 2f_{n-2})$$

Exercício 10.14 *Resolva o sistema*

$$\begin{cases} x_1' = \text{sen}x_1 + \cos(tx_2) \\ x_2' = \text{sen}(tx_1)t^{-1} \end{cases}, \; x_1(-1) = 2,37, \; x_2(-1) = -3,48$$

usando o método de Euler, para $-1 \leq t \leq 1$ *e* $h = 0,01$.

Exercício 10.15 *Seja o sistema*

$$\begin{cases} x_1' = (-1 - 9c^2 + 12sc)x_1 + (12c^2 + 9sc)x_2 \\ x_2' = (-12s^2 + 9s)x_1 + (-1 - 9s^2 - 12sc)x_2 \end{cases}, \; x_1(0) = -2, \; x_2(0) = 1$$

onde $c = \cos(6t)$, $s = \text{sen}(6t)$. *Para* $0 \leq t \leq 10$, $h = 0,01$, *responda:*

1. *Compare a solução numérica com a solução analítica,*

$$\begin{cases} x_1 = e^{-13t}(s - 2c) \\ x_2 = e^{-13t}(2s + c) \end{cases}$$

2. *Recalcule a solução numérica, para* $-0,01 \leq t \leq 10$ *e* $0,02 \leq t \leq 10$. *Houve diferença da solução obtida no item 1? Explique.*

Exercício 10.16 *Uma das equações básicas dos circuitos elétricos é*

$$L\frac{di}{dt} + Ri = E$$

onde L *é a indutância,* R *é a resistência,* i *é a corrente e* E, *a força eletromotriz. Considere* $L = 3\,H$, $R = 15\,\Omega$, $E = 110\,V$ *e* $i = 0$ *quando* $t = 0$. *Determine o valor da corrente quando* $t = 0,5\,s$, *tomando* $h = 0,1\,s$ *e usando o método de predição-correção de quarta ordem.*

Exercício 10.17 *Um exemplo de um sistema de equações diferenciais não lineares é o modelo presa-predador. Seja* $x(t)$ *a população de coelhos no tempo* t *e* $y(t)$ *a de raposas. O modelo presa-predador exige que* $x(t)$ *e* $y(t)$ *satisfaçam*

$$\begin{aligned} x' &= Ax - Bxy \\ y' &= Cxy - Dy \end{aligned}$$

Para fins de simulação numérica, pode-se considerar os coeficientes:

$$A = 2 \qquad B = 0,02 \qquad C = 0,0002 \qquad D = 0,8,$$

Use o método de Runge-Kutta para resolver a equação diferencial no intervalo $[0,5]$ *se*

1. $x(0) = 3000$ *coelhos e* $y(0) = 120$ *raposas;*
2. $x(0) = 5000$ *coelhos e* $y(0) = 100$ *raposas.*

Exercício 10.18 *Resolva o PVF linear*

$$\begin{cases} x'' = e^{t-3} + (t^2 + 2)x + (\text{sen}t)x' \\ x(2,6) = 7, \; x(5,1) = -3 \end{cases}$$

usando as equações (10.80)-(10.84). O PVI associado deverá ser resolvido pelo método de Runge-Kutta-Fehlberg.

Capítulo 11

Solução Numérica de Equações Diferenciais Parciais

11.1 Introdução

Uma equações diferencial parcial (EDP) pode ser escrita na forma geral

$$a\frac{\partial^2\phi}{\partial x^2} + b\frac{\partial^2\phi}{\partial x\partial y} + c\frac{\partial^2\phi}{\partial y^2} + d\frac{\partial\phi}{\partial x} + e\frac{\partial\phi}{\partial y} + f\phi + g = 0 \tag{11.1}$$

onde a, b, c, d, e, f e g podem ser funções das variáveis independentes x e y e da variável dependente ϕ, em uma região no plano $\mathbf{R}^2$, em coordenadas cartesianas.

As EDPs podem ser classificadas em *elípticas*, *parabólicas* ou *hiperbólicas*, dependendo do valor de $b^2 - 4ac$ ser negativo, zero ou positivo, respectivamente.

Como exemplos de EDPs elípticas podemos citar a *equação de Poisson*

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = g \tag{11.2}$$

e de *Laplace*

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \tag{11.3}$$

as quais são associadas geralmente a problemas em equilíbrio. Uma maneira de se expressar o potencial de velocidade de um fluido incompressível, não-viscoso, em regime estável é através da equação de Laplace: a taxa com a qual tal fluido entra em uma determinada região é igual àquela com a qual ele sai.

Já na teoria eletromagnética, o teorema de Gauss nos diz que o fluxo elétrico que passa através de uma superfície fechada é igual à carga total dentro da superfície; isto pode ser expresso por uma equação de Poisson,

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\rho}{\epsilon} = 0 \tag{11.4}$$

onde V é o potencial elétrico associado a uma distribuição bi-dimensional de carga de densidade ρ e ϵ é a constante dielétrica.

Até hoje, apenas um número limitado de equações elípticas foram resolvidas analiticamente, com sua utilidade restrita a casos onde a região de estudo considerada tem uma forma geométrica simples. O mesmo pode ser dito de equações parabólicas e hiperbólicas. Por essa razão, a solução dessas equações é feita, essencialmente, de forma numérica, com métodos específicos para cada tipo de EDP.

11.2 Equações parabólicas

Começaremos investigando a solução numérica de EDPs parabólicas com um dos mais simples exemplos: a equação adimensional

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \tag{11.5}$$

a qual expressa a distribuição de temperatura U em uma barra isolada termicamente ao longo de seu comprimento, x, t segundos após ter sido aquecida (ou resfriada). Em tal problema, as temperaturas nos dois extremos da barra são conhecidas ao longo do tempo – ou seja, as *condições de fronteira* são conhecidas. É também usual conhecer a distribuição de temperatura na barra em um certo instante, o qual é chamado de *tempo zero*; essa distribuição é chamada de *condição inicial*.

Vejamos, então, um diagrama que explica o processo de integração a ser efetuado (ver [14]). Suponha uma barra de comprimento l, C uma curva de fronteira e S a região englobada por essa curva, conforme a figura 11.1. Veja que C é uma curva aberta no plano $x-t$; e a área S limitada

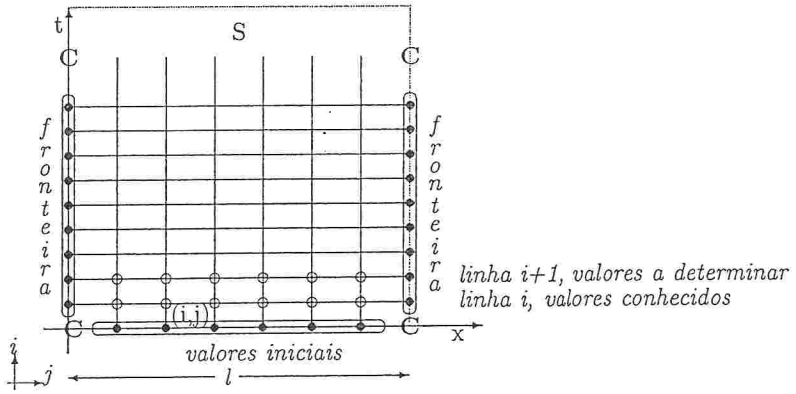

Figura 11.1: *Malha de pontos para uma EDP parabólica.*

por $0 \leq t < \infty$, $0 \leq x \leq l$. O processo de integração em S através de diferenças-finitas consiste em colocar-se uma malha com espaçamento k e h nas direções t e x respectivamente, e aproximar as derivadas em cada um dos pontos de intersecção nessa malha.

11.2.1 Método explícito

Para a equação (11.5), aproximaremos as derivadas de primeira e de segunda ordem através de diferenças-finitas (vide seção 2.5), i.e.

$$\frac{\partial u}{\partial t} \approx \frac{u_{i+1,j} - u_{i,j}}{k}$$

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i,j+1}}{h^2}$$

onde k e h são os espaçamentos nas direções t e x, respectivamente. Substituindo essas aproximações em (11.5), vem

$$\frac{u_{i+1,j} - u_{i,j}}{k} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i,j+1}}{h^2}$$

e, isolando o termo $u_{i+1,j}$, obtemos

$$u_{i+1,j} = ru_{i,j-1} + (1-2r)u_{i,j} + ru_{i,j+1}, \quad r = \frac{k}{h^2} \tag{11.6}$$

a qual nos dá a temperatura U em cada ponto j no $(i+1)$-ésimo tempo. Note que os pontos discretos são $x_j = jh$ e $t_i = ik$. Os exemplos a seguir mostram como resolver problemas com base na equação (11.6).

Exemplo 11.1 *Suponha uma barra de metal isolada termicamente, com as suas duas extremidades em contato com blocos de gelo a $0°C$, e aquecida instantaneamente em seu ponto médio por um maçarico. Qual a temperatura da barra após um certo tempo?*

Solução: *Note que, como a barra permanece em contato com gelo durante toda a simulação, devemos esperar que, após ter sido instantaneamente aquecida, sua temperatura deverá cair até $0°C$, depois de um certo tempo.*

Suponha que a distribuição inicial de temperatura na barra seja dada por

$$u = 2x, \quad 0 \leq x \leq \frac{1}{2}; \quad u = 2(1-x), \quad \frac{1}{2} \leq x \leq 1$$

Pela formulação do problema, vemos que os extremos da barra estão a $0°C$. Então, podemos especificar as condições iniciais *(CI) e* de fronteira *(CF) como:*

$$CI \; : \; \begin{cases} u = 2x, \; 0 \leq x \leq \frac{1}{2} \\ u = 2(1-x), \; \frac{1}{2} \leq x \leq 1 \end{cases}, \quad t = 0$$

$$CF \; : \; \begin{cases} u = 0, \; x = 0 \\ u = 0, \; x = 1 \end{cases}, \quad t > 0$$

Dividamos então a barra em dez pedaços e integremos a equação (11.5) em mil passos, i.e., tomemos $h = 1/10$ e $k = 1/1000$, tal que $r = k/h^2 = 1/10$. A equação (11.6) pode ser simplificada e escrita como

$$u_{i+1,j} = \frac{1}{10}(u_{i,j-1} + 8u_{i,j} + u_{i,j+1}) \tag{11.7}$$

e, identificando as variáveis envolvidas na malha, vemos que o cálculo dos $u_{i+1,j}$ corresponde à seguinte "molécula", onde os números dentro de cada "átomo" são os fatores multiplicadores de u nos pontos da malha, indicados ao lado de cada "átomo":

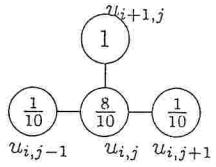

Figura 11.2: *"Molécula" computacional para o método explícito, $r = 1/10$.*

Agora, aplicando a equação (11.7), obtemos alguns valores, mostrados na tabela 11.1.

x	0,1	0,2	0,3	0,4	0,4	0,5	0,6
$t = 0$	0	0,2	0,4	0,6	0,8	1,0	0,8
$t = 0,001$	0	0,2	0,4	0,6	0,8	0,96	0,8
$t = 0,002$	0	0,2	0,4	0,6	0,7960	0,9280	0,7960
$t = 0,003$	0	0,2	0,4	0,5996	0,7896	0,9016	0,7896

Tabela 11.1: *Alguns valores para a equação (11.7), com $r = 1/10$.*

Considerando que a solução analítica para o problema, sujeita àquelas CI e CF, é dada por

$$u(x,t) = \frac{8}{\pi^2} \sum_{i=1}^{\infty} \frac{1}{n^2} \left(\text{sen}\frac{n\pi}{2}\right) (\text{sen} n\pi x) e^{-n^2\pi^2 t}$$

podemos calcular os erros correspondentes à solução numérica para alguns pontos, conforme mostra a tabela 11.2. Nota-se que, para $x = 0,3$, o erro é pequeno; já para $x = 0,5$, o erro é maior,

$x = 0,3$	t	numérica	analítica	erro (%)
	0,005	0,5971	0,5966	0,08
	0,01	0,5822	0,5799	0,4
	0,02	0,5373	0,5334	0,7
	0,1	0,2472	0,2444	1,1
$x = 0,5$				
	0,005	0,8597	0,8404	2,3
	0,01	0,7867	7743	1,6
	0,02	0,6891	0,6809	1,2
	0,1	0,3056	0,3021	1,2

Tabela 11.2: *Erros na aproximação numérica da equação (11.7), com $r = 1/10$.*

particularmente devido às CI, pois

$$\left.\frac{\partial u}{\partial x}\right|_{x=\frac{1}{2}^-} = 2 \neq -2 = \left.\frac{\partial u}{\partial x}\right|_{x=\frac{1}{2}^+}.$$

No entanto, à medida que o processo de integração prossegue, o erro diminui.

Richtmeyer e Morton (1967) mostraram que, para o esquema em diferenças-finitas utilizado no exemplo 11.1, se a função inicial e suas $p-1$ primeiras derivadas são contínuas, e a p-ésima derivada é discontínua de forma ordinária (i.e., varia em saltos finitos), então, para k pequeno, o erro é menor ou igual a

$$k^{\frac{p+2}{p+4}}$$

No exemplo 11.1, $p = 1$, então o erro é $k^{\frac{3}{5}} = 0,016$; comparando com os valores presentes na tabela 11.2, vemos que essa condição é satisfeita.

Exemplo 11.2 *Para o mesmo problema no exemplo 11.1, utilize $h = 1/10$ e $k = 5/1000$.*
Solução: *Nesse caso, $r = 1/2$. A equação (11.6) reduz-se a*

$$u_{i+1,j} = \frac{1}{2}(u_{i,j-1} + u_{i,j+1}) \tag{11.8}$$

Alguns dos valores calculados são mostrados na tabela 11.3; os erros em $x = 0,3$ são mostrados na tabela 11.4. Comparando com os dados na tabela 11.2, vemos que os erros aumentaram; isso

x	0,1	0,2	0,3	0,4	0,4	0,5	0,6
$t = 0$	0	0,2	0,4	0,6	0,8	1,0	0,8
$t = 0,005$	0	0,2	0,4	0,6	0,8	0,8	0,8
$t = 0,010$	0	0,2	0,4	0,6	0,7	0,8	0,7
$t = 0,015$	0	0,2	0,4	0,55	0,7	0,7	0,7

Tabela 11.3: *Alguns valores para a equação (11.8), com $r = 1/2$.*

nos leva a supor que o valor de k está intimamente ligado ao erro existente na solução.

Exemplo 11.3 *Para o mesmo problema no exemplo 11.1, utilize $h = 1/10$ e $k = 1/100$.*
Solução: *Nesse caso, $r = 1$. A equação (11.6) reduz-se a*

$$u_{i+1,j} = u_{i,j-1} - u_{i,j} + u_{i,j+1} \tag{11.9}$$

A tabela 11.5 mostra alguns dos valores calculados os quais, obviamente, não representam o comportamento físico esperado.

$x = 0,3$	t	numérica	analítica	erro (%)
	0,005	0,6	0,5966	0,57
	0,01	0,6	0,5799	3,5
	0,02	0,55	0,5334	3,1
	0,1	0,2484	0,2444	1,6

Tabela 11.4: *Erros na aproximação numérica da equação (11.8), com* $r = 1/2$.

x	0,1	0,2	0,3	0,4	0,4	0,5	0,6
$t = 0$	0	0,2	0,4	0,6	0,8	1,0	0,8
$t = 0,01$	0	0,2	0,4	0,6	0,8	0,6	0,8
$t = 0,02$	0	0,2	0,4	0,6	0,4	1,0	0,4
$t = 0,03$	0	0,2	0,4	0,2	1,2	−0,2	1,2
$t = 0,04$	0	0,2	0,0	1,4	−1,2	2,6	−1,2

Tabela 11.5: *Alguns valores para a equação (11.9), com* $r = 1$.

O método explícito expresso pela equação (11.6) é computacionalmente simples, porém ele apresenta uma restrição. O passo de integração temporal, k, deve ser necessariamente pequeno, uma vez que o processo de integração só válido quando

$$k \geq \frac{h^2}{2} \tag{11.10}$$

Além disso, o particionamento em x deve ser suficientemente grande – levando a um h pequeno – para que as aproximações das derivadas espaciais, usando diferenças-finitas, sejam aceitáveis. No exemplo 11.3, a condição (11.10) foi violada, levando a resultados incorretos (do ponto de vista físico).

A fim de remover essa restrição no passo de integração temporal, devemos recorrer a um outro método, conforme descrito a seguir.

11.2.2 Método de Crank-Nicolson

Em 1947, Crank e Nicolson propuseram um método válido para quaisquer valores finitos de r. Eles consideraram que a EDP (11.5 é satisfeita no ponto médio $\left(i + \frac{1}{2}k, jh\right)$, ou seja,

$$\left(\frac{\partial u}{\partial t}\right)_{i+\frac{1}{2},j} = \left(\frac{\partial^2 u}{\partial x^2}\right)_{i+\frac{1}{2},j}$$

e, substituindo estas derivadas pelas expressões em diferenças finitas já vistas, temos

$$\frac{u_{i+1,j} - u_{i,j}}{k} = \frac{1}{2}\left(\frac{u_{i+1,j-1} - 2u_{i+1,j} + u_{i+1,j+1}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2}\right)$$

ou

$$-ru_{i+1,j-1} + (2 + 2r)u_{i+1,j} - ru_{i+1,j} = ru_{i,j-1} + (2 - 2r)u_{i,j} + ru_{i,j+1} \tag{11.11}$$

onde $r = k/h^2$. Na equação (11.11), temos três termos conhecidos e três a determinar, os quais referem-se ao tempo $i + 1$; daí, o método de Crank-Nicolson é dito ser *implícito*.

Se cada linha da malha tiver n pontos, então (11.11) é um sistema de equações lineares de n equações a n variáveis, onde o termo independente de cada sistema, onde calculamos u no tempo $i + 1$, é composto pelos valores de u no tempo i. Note que a resolução do sistema de equações deve ser efetuada *a cada passo de integração no tempo*. O exemplo a seguir mostra como utilizar o método.

Exemplo 11.4 *Resolva o problema no exemplo 11.1 com o método de Crank-Nicolson, usando* $h = 1/10$.

Solução: *Ao selecionarmos* k, *devemos verificar que uma escolha cuidadosa de* r *permite simplificar a equação (11.11); p.ex., com* $r = 1$, *o termo* $u_{i,j}$ *é removido. Para esse valor de* r, *temos* $k = 1/100$. *Então, a equação governante, nesse caso, é*

$$-u_{i+1,j-1} + 4u_{i+1,j} - u_{i+1,j+1} = u_{i,j-1} + u_{i,j+1}$$

cuja "molécula" computacional é mostrada na figura 11.3. Devido à simetria do problema, em

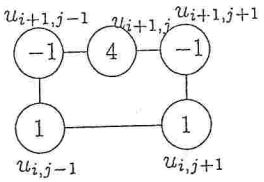

Figura 11.3: *"Molécula" computacional para o método de Crank-Nicolson,* $r = 1$.

relação ao quinto ponto, $x = 1/2$, *conforme mostrado na figura 11.4, podemos reduzir o esforço computacional necessário; basta obter os valores de* $u_{i+1,1}$, $u_{i+1,2}$, $u_{i+1,3}$, $u_{i+1,4}$ *e* $u_{i+1,5}$. *O*

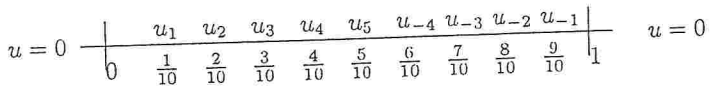

Figura 11.4: *Disposição dos pontos na malha.*

sistema de equações correspondente pode ser escrito na forma matricial como:

$$\begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & -1 & 4 & -1 & \\ & & -1 & 4 & -1 \\ & & & -2 & 4 \end{bmatrix} \begin{bmatrix} u_{i+1,1} \\ u_{i+1,2} \\ u_{i+1,3} \\ u_{i+1,4} \\ u_{i+1,5} \end{bmatrix} = \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ u_{i,3} \\ u_{i,4} \\ u_{i,5} \end{bmatrix}$$

cuja matriz de coeficientes é levemente não-simétrica, apresentando dominância diagonal.

11.2.2.1 Aproximação ponderada

Se considerarmos que a EDP (11.5) é satisfeita, agora, num ponto $(i + \theta k, jh)$, $0 \leq \theta \leq 1$, obtemos uma generalização do método de Crank-Nicolson:

$$\frac{u_{i+1,j} - u_{i,j}}{k} = \frac{1}{h^2}\left(\theta(u_{i+1,j-1} - 2u_{i+1,j} + u_{i+1,j+1}) + (1-\theta)(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})\right) \quad (11.12)$$

A equação (11.12) corresponde ao método explícito, para $\theta = 0$; ao de Crank-Nicolson, para $\theta = 1/2$; e a um método completamente implícito, para $\theta = 1$. Para $1/2 \leq \theta \leq 1$, (11.12) é estável para qualquer valor de r; para $0 \leq \theta \leq 1/2$, devemos ter $r \leq (2(1-2\theta))^{-1}$.

11.2.3 Condições de fronteira

O principal problema, ao se resolver uma equação diferencial parcial, é o tratamento adequado das condições de fronteira.

Condições simples, como as de Dirichlet ($u = 0$), são facilmente tratadas. Por exemplo, considere a aproximação central por diferenças finitas da derivada $\partial u/\partial x$,

$$\frac{u_{j+1} - u_{j-1}}{2h}.$$

Se a condição é $u = 0$ em $x = 0$, então para o ponto na malha em $j = 0$, $u_0 = 0$. Daí, a aproximação para a derivada no primeiro ponto, $x = h$, i.e. $j = 1$, passa a ser

$$\frac{u_1 - 0}{2h}$$

O mesmo procedimento é válido para outras aproximações em diferenças finitas, como visto nos exemplos 11.1-11.3.

Condições de fronteira envolvendo *derivadas* ocorrem com freqüência, normalmente para indicar a inexistência de fluxo naquela parte da região em estudo, ou a taxa com a qual uma dada quantidade varia. Por exemplo, quando a superfície de um objeto condutor de calor é termicamente isolado, podemos dizer que

$$\frac{\partial U}{\partial n} = 0$$

i.e., a derivada de U na direção *normal* à superfície (indicada por n) é nula.

Um outro exemplo refere-se à transmissão de calor. A taxa com que o calor é transferido por irradiação de uma superfície quente para o meio ao seu redor, e uma temperatura v, é normalmente assumida como proporcional a $U - v$. Como na teoria de condução de calor a premissa fundamental é de que o calor que flui através de uma superfície é igual a $-\kappa\frac{\partial U}{\partial n}$ unidades de calor por unidade de tempo na direção normal à superfície, podemos escrever a condição de fronteira como

$$-\kappa\frac{\partial U}{\partial n} = H(U - v)$$

onde κ é a constante de condutividade térmica do material e H é a constante de transferência de calor da superfície. O sinal é tomado de forma a indicar que o fluxo de calor é na direção oposta àquela em que U cresce algebricamente. Podemos, então, simplificar a condição de fronteira acima para

$$\frac{\partial U}{\partial n} = -s(U - v), \quad s > 0,$$

Considere, então, uma barra fina, termicamente isolada, e que irradia calor no seu extremo $x = 0$. A temperatura nesse extremo, em $t = 0$, é, portanto, desconhecida, e necessitamos de uma outra equação; essa pode ser a própria condição de fronteira, se usarmos uma aproximação frontal para a derivada em U, pois

$$-\frac{\partial U}{\partial x} = -s(U - v)$$

será aproximado por

$$\frac{u_{i,1} - u_{i,0}}{h} = s(u_{i,0} - v)$$

Um sinal negativo deve ser associado à derivada pois a normal, apontando para fora da barra, em $x = 0$, tem a direção $-x$.

Se desejarmos aproximar a derivada $\partial U/\partial x$ por uma aproximação central, então é necessário introduzir uma célula fictícia $u_{i,-1}$, imaginando-se que a barra estende-se até $-h$, conforme a figura 11.5. Podemos, então, escrever

$$\frac{u_{i,1} - u_{i,-1}}{2h} = s(u_{i,0} - v)$$

porém, agora, devemos eliminar $u_{i,-1}$, já que ele é fictício, conforme mostra o exemplo a seguir.

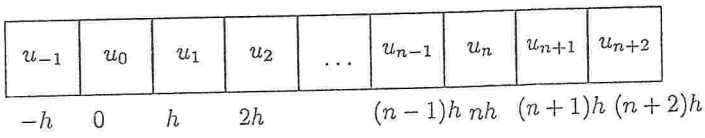

Figura 11.5: *Malha com pontos fictícios.*

Exemplo 11.5 *Seja uma barra fina, termicamente isolada, a uma temperatura diferente da temperatura ambiente, a qual irradia calor através das suas duas extremidades. A EDP é*

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial t^2}$$

sujeita às

$$\begin{aligned} CI &: U = 1, \quad 0 \leq x \leq 1, \quad t = 0 \\ CF &: \begin{cases} \frac{\partial U}{\partial x} = U, & x = 0, \quad t > 0 \\ \frac{\partial U}{\partial x} = -U, & x = 1, \quad t > 0 \end{cases} \end{aligned}$$

Obtenha as equações governantes de acordo com o método explícito.

Solução: *Usando a equação (11.6), podemos escrever, em $x = 0$,*

$$u_{i+1,0} = u_{i,0} + r(u_{i,-1} - 2u_{i,0} + u_{i,1})$$

A CF em $x = 0$, usando uma aproximação central, é

$$\frac{u_{i,1} - u_{i,-1}}{2h} = u_{i,0}$$

de onde $u_{i,-1} = u_{i,1} - 2hu_{i,0}$; substituindo na equação para $u_{i,0}$, temos

$$\begin{aligned} u_{i+1,0} &= u_{i,0} + r(u_{i,1} - 2hu_{i,0} - 2u_{i,0} + u_{i,1}) = \\ &= u_{i,0} = 2r(u_{i,1} - (1+h)u_{i,0}) \end{aligned}$$

De forma similar, em $x = 1$, temos como CF

$$\frac{u_{i,n+2} - u_{i,n}}{2h} = u_{i,n+1}$$

e, como a equação (11.5) é expressa no ponto $x = 1$ como

$$u_{i+1,n+1} = u_{i,n+1} + r(u_{i,n} - 2u_{i,n+1} + u_{i,n+2})$$

podemos eliminar $u_{i,n+2}$, tal que

$$u_{i+1,n+1} = u_{i,n+1} + 2r(u_{i,n} - (1+h)u_{i,n+1})$$

Para resolver esse problema, então, vamos utilizar três equações: uma para o ponto $u_{i+1,0}$, outra para o ponto $u_{i+1,n+1}$ e a equação (11.6) para os pontos $u_{i+1,j}$, $1 \leq j \leq n$.

11.3 Equações diferenciais parciais elípticas

As EDPs *elípticas* são normalmente relacionadas a problemas em equilíbrio e as suas soluções representam um máximo ou mínimo da integral que representa a energia do sistema. As mais conhecidas, e importantes são as equações de Poisson (11.2) e de Laplace (11.3). Suas aplicações são as mais variadas: p.ex., a equação de Poisson representa o movimento de um fluido viscoso incompressível, a baixa velocidade; a equação de Laplace é empregada para descrever o potencial eletromagnético, dentre outras.

O intervalo de integração de uma EDP elíptica é sempre uma área S cercada por uma curva *fechada* C. As condições de fronteira especificam ou o valor da função ou de sua derivada em cada ponto de C; é comum, também, que em certas regiões de C seja especificado o valor da função e, noutras, o da sua derivada.

A solução de uma EDP elíptica através de sua discretização em diferenças finitas leva à solução de um sistema de equações lineares, tipicamente grande e esparso, para os quais os métodos iterativos, apresentados no Capítulo 4, são particularmente indicados.

Considere a equação

$$\nabla^2 u = f(x, y) \tag{11.13}$$

na região $R = [0,1] \times [0,1]$ (i.e., o quadrado unitário), sujeita às condições de fronteira $u = 0$ em ΩR.

Discretizamos, então, a região com uma malha cartesiana com espaçamento $h = 1/m$, idêntico nas direções x e y, conforme o diagrama da figura 11.6. Note que temos $n = m - 1$ pontos ao

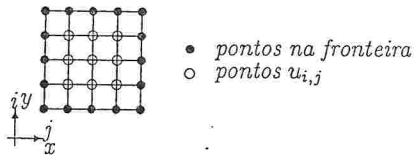

Figura 11.6: *Malha de pontos para uma EDP elíptica.*

longo de cada direção, totalizando n^2 pontos $u_{i,j}$ onde obteremos a aproximação para a EDP.

Aproximando as derivadas parciais de segunda ordem por diferenças finitas, temos

$$\frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} + \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} = f(jh, ih), \quad 0 < i, j < m \tag{11.14}$$

onde $jh = x_j$ e $ih = y_i$. Usando um ordenamento natural dos pontos $u_{i,j}$ na malha, ao longo das linhas verticais, obtemos um número $k = (i-1)n + j$, tal que $u_{i,j} \equiv u_k$, conforme mostrado na figura 11.7. O ordenamento natural ao longo das linhas horizontais é equivalente: nesse caso, teríamos $k = (j-1)n + i$. Avaliando a equação (11.14) em cada um dos pontos i, j, obteremos

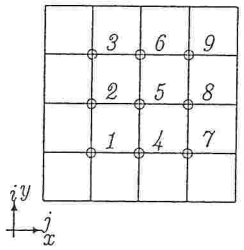

Figura 11.7: *Ordenação natural dos pontos na malha.*

um sistema de equações lineares, conforme mostra o exemplo abaixo.

Exemplo 11.6 *Calcule u que satisfaça a EDP*

$$-\nabla^2 u = -2(x^2 + y^2)$$

na região $R = [0,1] \times [0,1]$, sujeita a $u = 0$ nas linhas $x = 0$ e $y = 0$; $u = y^2$ na linha $x = 1$ e $u = x^2$ na linha $y = 1$.

Solução: *Apenas para fins de explanação, vamos utilizar uma malha quadrada com $m = 4$, mas cabe ressaltar que, tipicamente, devemos utilizar m 100. Dessa forma, estaremos avaliando*

a EDP acima em $n^2 = (m-1)^2 = 9$ *pontos, o que nos levará a um sistema de equações lineares de* 9 *equações a* 9 *variáveis. O espaçamento é* $h = 1/4$.

Escrevendo, então, a equação (11.14) para cada um dos pontos, utilizando o ordenamento natural conforme mostrado na figura 11.7, temos:

$$\begin{aligned}
1 &: (0 + 2u_1 - u_4) + (0 + 2u_1 - u_2) = -2h^2(h^2 + h^2)\\
2 &: (0 + 2u_2 - u_5) + (u_1 + 2u_2 - u_3) = -2h^2(h^2 + 4h^2)\\
3 &: (0 + 2u_3 - u_6) + (u_2 + 2u_3 - h^2) = -2h^2(h^2 + 9h^2)\\
4 &: (u_1 + 2u_4 - u_7) + (0 + 2u_4 - u_5) = -2h^2(4h^2 + h^2)\\
5 &: (u_2 + 2u_5 - u_8) + (u_4 + 2u_5 - u_6) = -2h^2(4h^2 + 4h^2)\\
6 &: (u_3 + 2u_6 - u_9) + (u_5 + 2u_6 - 4h^2) = -2h^2(4h^2 + 9h^2)\\
7 &: (u_4 + 2u_7 - h^2) + (0 + 2u_7 - u_8) = -2h^2(9h^2 + h^2)\\
8 &: (u_5 + 2u_8 - 4h^2) + (u_7 + 2u_8 - u_9) = -2h^2(9h^2 + 4h^2)\\
9 &: (u_6 + 2u_9 - 9h^2) + (u_8 + 2u_9 - 9h^2) = -2h^2(9h^2 + 9h^2)
\end{aligned}$$

onde os valores 0 *correspondem àqueles pontos na fronteira, pois* $u = 0$ *é a condição de fronteira. Multiplicando por* -1 *todas as equações e combinando os termos, obtemos o seguinte sistema de equações, na forma matricial:*

$$\begin{bmatrix}
4 & -1 & & -1 & & & & & \\
-1 & 4 & -1 & & -1 & & & & \\
 & -1 & 4 & & & -1 & & & \\
-1 & & & 4 & -1 & & -1 & & \\
 & -1 & & -1 & 4 & -1 & & -1 & \\
 & & -1 & & -1 & 4 & & & -1 \\
 & & & -1 & & & 4 & -1 & \\
 & & & & -1 & & -1 & 4 & -1 \\
 & & & & & -1 & & -4 & 4
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix}
=
\begin{bmatrix} -0,0156 \\ -0,0391 \\ -0,0156 \\ -0,0391 \\ -0,0625 \\ 0,1484 \\ -0,0156 \\ 0,1484 \\ 0,9144 \end{bmatrix}
\qquad (11.15)$$

cuja solução é

$$u = (0,0039, 0,0156, 0,0352, 0,0156, 0,0625, 0,1406, 0,0352, 0,1406, 0,3164)$$

Note que, pela definição do problema, temos $u = x^2y^2$; *podemos confirmar que os valores obtidos como solução do sistema (11.15) satisfaz a EDP, como mostra a tabela 11.6.*

k	x	y	$u = x^2y^2$	u_k	erro relativo
1	0,25	0,25	0,0039	0,0039	$0,6661 \times 10^{-15}$
2	0,25	0,50	0,0156	0,0156	$0,4441 \times 10^{-15}$
3	0,25	0,75	0,0352	0,0352	0
4	0,50	0,25	0,0156	0,0156	$0,2220 \times 10^{-15}$
5	0,50	0,50	0,0625	0,0625	$0,1110 \times 10^{-15}$
6	0,50	0,75	0,1406	0,1406	0
7	0,75	0,25	0,0352	0,0352	0
8	0,75	0,50	0,1406	0,1406	$0,1974 \times 10^{-15}$
9	0,75	0,75	0,3164	0,3164	0

Tabela 11.6: *Comparação entre os valores calculados para u pela solução direta do sistema e a solução exata.*

Como as matrizes que surgem da discretização em diferenças finitas de uma EDP são esparsas, é comum se utilizar métodos iterativos na resolução do sistema de equações lineares, como mostra o exemplo a seguir.

Exemplo 11.7 *Resolvendo o sistema 11.15 através do método dos Gradientes-Conjugados (vide seção 4.7 e algoritmo 4.7.1), a uma tolerância de* 10^{-10}*, obtém-se como solução*

$$u = (0,0039, 0,0156, 0,0352, 0,0156, 0,0625, 0,1406, 0,0352, 0,1406, 0,3164)$$

A tabela 11.7 mostra que os erros relativos dessa solução também são aceitáveis.

k	x	y	$u = x^2y^2$	u_k	erro relativo
1	0,25	0,25	0,0039	0,0039	$0,1332 \times 10^{-14}$
2	0,25	0,50	0,0156	0,0156	$0,0555 \times 10^{-14}$
3	0,25	0,75	0,0352	0,0352	$0,0197 \times 10^{-14}$
4	0,50	0,25	0,0156	0,0156	$0,0555 \times 10^{-14}$
5	0,50	0,50	0,0625	0,0625	$0,0222 \times 10^{-14}$
6	0,50	0,75	0,1406	0,1406	$0,0197 \times 10^{-14}$
7	0,75	0,25	0,0352	0,0352	$0,0395 \times 10^{-14}$
8	0,75	0,50	0,1406	0,1406	$0,0197 \times 10^{-14}$
9	0,75	0,75	0,3164	0,3164	$0,0175 \times 10^{-14}$

Tabela 11.7: *Comparação entre os valores calculados para u pela solução iterativa do sistema, através do método dos Gradientes-Conjugados, e a solução exata.*

Assim como nas equações parabólicas, pode-se especificar condições de Neumann (envolvendo derivadas) na fronteira.

11.4 Exercícios

Exercício 11.1 *Considere a equação*

$$\frac{\partial U}{\partial t} = \kappa \frac{\partial^2 U}{\partial x^2}, \quad \kappa < 1$$

Estabeleça a condição necessária para convergência de um método explícito e mostre o que acontece quando $\kappa = 0,5$ *e* $\kappa = 10^{-6}$.

Exercício 11.2 *Calcule a solução aproximada de*

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \cos(y)\frac{\partial U}{\partial x} - \text{sen}(x)\frac{\partial U}{\partial y} = 0$$

no retângulo unitário, com condições de Dirichlet na fronteira. Resolva o problema através de uma aproximação em diferenças-finitas, utilizando diferenças ascendentes para as derivadas de primeira ordem.

Exercício 11.3 *Resolva o exercício 11.2 utilizando diferenças centrais para as derivadas de primeira ordem. Explique o que ocorre.*

Bibliografia

[1] M. Abramowitz and J.A. Stegun. *Handbook of Mathematical Functions.* Dover, New York, 1964.

[2] F.S. Acton. *Numerical Methods That Work.* Mathematical Association of America, Washington, 1990.

[3] D.M. Cláudio. *Cálculo numérico computacional : teoria e prática.* Atlas, São Paulo, 1989.

[4] D.M. Cláudio and J.A. Royo dos Santos. *Microcomputadores e Minicalculadoras - seu uso em Ciências e Engenharia.* Edgard Blücher, São Paulo, 1983.

[5] J.E. Dennis Jr. and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, 1984.

[6] A.R. Gourlay and G.A. Watson. *Computational Methods for Matrix Eigenproblems.* J. Wiley & Sons, New York, 1973.

[7] P. Henrici. *Applied and computational complex analysis, V.1.* J. Wiley & Sons, New York, 1988.

[8] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1 of *Wiley Series in Numerical Methods in Engineering.* J. Wiley & Sons, New York, 1990.

[9] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* SIAM, Philadelphia, 1995.

[10] D.R. Kincaid and W. Cheney. *Numerical Analysis.* Brooks/Cole, Pacific Grove, 1991.

[11] U.S. General Accounting Office. Patriot Missile Defense: Software Problem Led to Failure at Dharan, Saudi Arabia. Report GAO/IMTEC-92-26, U.S. GAO, 1992.

[12] M.L. Overton. *Numerical Computing with IEEE Floating Point Arithmetic.* SIAM, Philadelphia, 2001.

[13] Youcef Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, Boston, 1995.

[14] G.D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods.* Oxford University Press, Oxford, 3rd edition, 1985.

Índice